ADQUISICIÓN DE INFORMACIÓN DE PROFUNDIDAD MEDIANTE LA TÉCNICA "STRUCTURED LIGHT, THREE PHASE-SHIFT"

LUIS FERNANDO AYUSO PÉREZ

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Sistemas Inteligentes

5 Septiembre de 2011

Directores:
Gonzalo Pajares Martinsanz (Dpto. Ingeniería del Software e Inteligencia Artificial)
Pedro Jesús Martín de la Calle (Dpto. Sistemas Informáticos y Computación)

Convocatoria: Septiembre 2011
Calificación: Sobresaliente

Autorización de difusión

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "ADQUISICIÓN DE INFORMACIÓN DE PROFUNDIDAD MEDIANTE LA TÉCNICA STRUCTURED LIGHT, THREE PHASE-SHIFTING", realizado durante el curso académico 2010-2011 bajo la dirección de Gonzalo Pajares Martinsanz en el Departamento de Ingeniería del Software e Inteligencia Artificial y Pedro Jesús Martín de la Calle en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

4 de Septiembre de 2011

Luis Fernando Ayuso Pérez

Resumen en castellano

En la última década se han realizado grandes avances en la captura de la estructura tridimensional de escenas u objetos reales. Las crecientes capacidades de cálculo hacen que hoy en día sea posible procesar volúmenes de datos anteriormente intratables, mejorando técnicas y haciendo éstas más rápidas y precisas. Esta capacidad de modelar dentro de un computador escenas del mundo real es muy ventajosa en áreas como la industria, el diseño industrial o las artes gráficas.

En esta memoria se puede encontrar una breve descripción de las técnicas existentes para la captación de información tridimensional, y entre éstas una vista general de las llamadas técnicas de luz estructurada. El tema central de este proyecto trata de estudiar una de las técnicas en auge, *three phase shift*, analizando su funcionamiento y puntuando las áreas de interés donde enfatizar esfuerzos. Se describe un análisis detallado de sus dos operaciones principales mediante el cual se evaluarán rendimiento y calidad, describiendo el modo de funcionamiento de los desarrollos actuales y la investigación llevada a cabo en estas dos etapas.

Todo el proyecto ha sido enfocado desde el punto de vista de la computación, estudiando el comportamiento de los algoritmos, en particular las etapas llamadas *phase wrap* y *phase unwrap*

Palabras clave

- Escaner 3D
- Luz estructurada
- Tres cambios de fase
- Mapa de profundidad
- *phase wrap*
- *phase unwrap*

Abstract

During last decade significant inprovement has being made capturing the three dimensional structure of real objects or scenes. Increasing computational capabilities now make it possible to process volumes of data previously intractable, improving technical and making them faster and more accurate. This ability to model inside a computer real-world scenes is very profitable in areas such as manufacturing, industrial design or graphic arts.

In this report you will find a brief overview of existing techniques for capturing three-dimensional information, and specifically a description of the so-called structured light techniques. This project aims to study one of the rising techniques, *three phase shift*, analyzing its performance and scoring areas of interest where more effort has to be done. A detailed analysis of its two main operations will be evaluated by performance and quality, describing the state of current developments and research conducted in this two stages.

The whole project has been approached from the point of view of computing, studying the behavior of algorithms, in particular stages called *phase wrap* and *phase unwrap*.

Keywords

- 3D scanner
- Structured Light
- Three phase shift
- Deep map
- Phase wrap
- Phase unwrap

Índice general

Índice I

Agradecimientos IV

1. Introducción 1
- 1.1. Escaneado 3D . . . 1
  - 1.1.1. Visión estereoscópica . . . 2
  - 1.1.2. Tiempo de vuelo . . . 3
  - 1.1.3. Luz estructurada . . . 3
- 1.2. Objetivos . . . 6
- 1.3. Organización de este trabajo . . . 6

2. Tres cambios de fase (Three phase shift) 7
- 2.1. Descripción del método . . . 7
- 2.2. Aplicaciones . . . 10
- 2.3. Problemas detectados . . . 10
  - 2.3.1. Problemas de captura . . . 11
  - 2.3.2. Problemas cromáticos . . . 11
  - 2.3.3. Problemas algorítmicos . . . 12

3. Phase wrap 14
- 3.1. Operación de *phase wrap* . . . 15
  - 3.1.1. Pre-requisitos . . . 15
- 3.2. Análisis de phase wrap . . . 15
  - 3.2.1. Aceleradores . . . 16
  - 3.2.2. Breve estudio sobre los filtros espaciales . . . 18
  - 3.2.3. Filtro Escogido . . . 21

4. Phase Unwrap 23
- 4.1. Operación básica . . . 23
- 4.2. Transformación unwrap . . . 24
- 4.3. Transformaciones en recorrido . . . 26
  - 4.3.1. Evaluación de la calidad . . . 28
  - 4.3.2. Algoritmo de inundación guiado por prioridad . . . 31

5. Resultados **34**
5.1. Resultados con respecto a *phase wrap* . . . 34
5.1.1. Características del mapa *wrap* que lo hacen inadecuado para el uso del filtro bilateral . . . 34
5.2. Consideraciones de rendimiento . . . 35
5.3. Resultados con respecto a *phase unwrap* . . . 36

6. Conclusión y trabajo futuro **39**
6.1. Conclusión . . . 39
6.2. Materiales . . . 39
6.3. Trabajo futuro . . . 41
6.3.1. Real-time framework . . . 41
6.3.2. GPGPU . . . 42

Bibliografia **46**

List of Figures **47**

A. Diagramas de ejecución **48**
A.1. *phase unwrap* Top-down . . . 48
A.2. *phase unwrap* por inundación . . . 48

Agradecimientos

Este proyecto no podría haber sido realizado sin la ayuda de Gonzalo Pajares, a quién agradezco la dedicación que me ha prestado durante la realización de éste y la redacción de la presente memoria.

También he de agradecer a Chus Martín y Antonio Gavilanes la confianza depositada, permitiéndome trabajar en un apasionante proyecto que ha despertado mi interés por la investigación y la ciencia.

Así mismo he de agradecer al proyecto CCG10-UCM/TIC-5476 que ha financiado la beca de investigación que he disfrutado durante el año 2011 y que me ha permitido implicarme como corresponde en el desarrollo de este trabajo.

Capítulo 1

Introducción

El método que se propone en este trabajo de investigación tiene como finalidad la obtención de la estructura tridimensional de la escena mediante la técnica conocida como Tres Cambios de Fase (en adelante *Three Phase Shift*). *Three Phase Shift* pertenece a un conjunto de técnicas llamadas Luz Estructurada. Pero ésta no es la única familia en el campo, es por esto que como paso previo al análisis se estudian las operaciones de captura de características tridimiensionales.

Posteriormente, se detalla la operación del método, dividiendo éste en sus etapas fundamentales a partir de las cuales se derivan los objetivos que se plantean en la investigación. También se describe la problemática derivada en cada una de ellas con base en las diferentes aplicaciones reales existentes, que en conjunto proporcionan las bases de la investigación desarrollada y justifican su estudio. Los restantes capítulos desarrollan en profundidad las diferentes etapas que conforman el método, así como los resultados obtenidos durante los experimentos propuestos.

1.1. Escaneado 3D

El reconocimiento de la profundidad en una escena escaneada por un dispositivo de captura de imágenes puede ser englobado dentro del ámbito de la visión por computador. La captura de las características tridimensionales es una de las características en las que se puede centrar un proceso de visión por computador. Las técnicas para capturar la geometría tridimensional de una escena y transferirla a un modelo matemático practicable por un computador son varias. Pueden ser catalogadas según su naturaleza en las tres categorías que se describen a continuación.

1.1.1. Visión estereoscópica

La Visión Estereoscópica por computador se basa en un juego de cámaras gemelas, las cuales tienen ejes ópticos paralelos. Al encontrarse las cámaras separadas en una línea horizontal, las imágenes conseguidas difieren solamente en la esta misma dirección. El algoritmo básico de correlación para la visión estereoscópica es el que se muestra en la figura 1.1. Mediante un origen virtual tridimensional (P), se realiza el trazado de líneas hacia los objetos (PO y PI), las cuales generan un plano epipolar, dentro del cual se debe encontrar la correspondencia de un mismo punto en el espacio real con sus dos representaciones en las imágenes de partida.

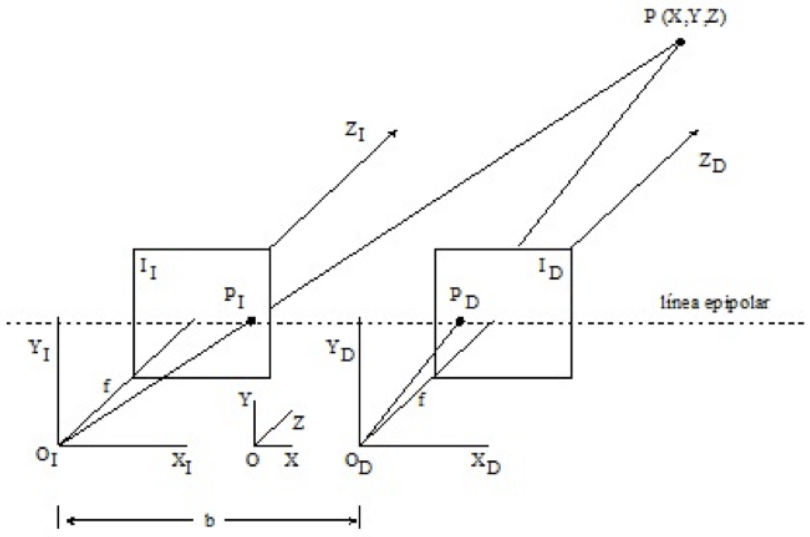


Figura 1.1: Modelo geométrico de un sistema de Visión Estereoscópica

La Visión Estereoscópica consta de ciertas etapas que la hacen diferente de las otras dos técnicas. En la figura 1.2 se puede observar como la mayor actividad es durante el procesado de la características de la escena presentes en la imagen, donde se buscan áreas de interés mediante resaltado de bordes, se diferencian regiones u objetos y se correlacionan en las dos imágenes. De esta manera se extrae la información de profundidad.

Lá técnica de luz estructurada tratada en este proyecto tiene un modo de proceder contrario a la visión estereoscópica, ya que no existe análisis de la escena. En la técnica de luz estructurada el proceso se realiza de forma idéntica para cada píxel de la imagen, ya que no existe diferencia entre píxeles que muestren determinados objetos o características particulares.

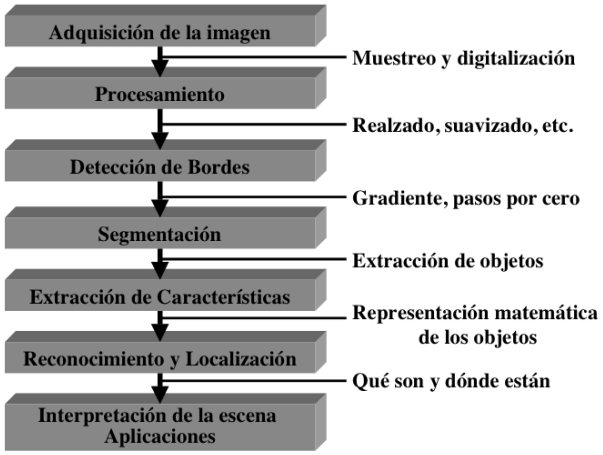


Figura 1.2: etapas en un proceso de Visión Estereoscópica

1.1.2. Tiempo de vuelo

Las técnicas de tiempo de vuelo se basan en sensores con una composición especial. Mientras que un sensor para la captura de imágenes normal consta de una matriz de celdas que proporcionan la imagen en color o escala de grises, una cámara de tiempo de vuelo consta de un sensor Photon Mixing Device (PMD), el cual adquiere de forma simultánea un valor de profundidad para el píxel y un valor de intensidad. Para que esto sea posible, es necesario que la escena esté iluminada, para lo cual se utiliza habitualmente luz infrarroja, ya que no interfiere con la iluminación de la escena y no altera el color de ésta.

Como ejemplo de dispositivo que implementa esta técnica se muestra en la figura 1.3 una estación de medida total Leica[5]. Estos aparatos robotizados hacen uso de una señal láser infrarroja para medir la distancia entre dos puntos separados varios kilómetros con precisión inferior al milímetro.

Esta técnica tiene una gran ventaja: gracias a la física utilizada la distancia medida es relativa a la distancia existente entre el sensor y la escena. Siendo el sensor el origen del sistema de referencia se puede considerar que las mediciones hechas son absolutas.

1.1.3. Luz estructurada

Las técnicas de luz estructurada hacen uso de un proyector para proyectar un patrón sobre la escena a escanear. Al hacer corresponder la imagen captada con el patrón original, es posible triangular la posición de cada píxel y determinar su profundidad con respecto al plano perpendicular a la cámara.

Dentro de la luz estructurada, existen diferentes técnicas dependiendo del tipo de patrón y del análisis posterior realizado:

Figura 1.3: Estación total de medida Leica

- **Códigos binarios**
  Los códigos binarios pueden conseguir buenos resultados con una técnica muy sencilla, sacrificando para ello el tiempo necesario de toma de varias imágenes con diferentes patrones.
  Como primer paso se proyecta la mitad del patrón en color blanco y la otra mitad en negro. Es posible entonces catalogar los píxeles en dos grupos. Posteriormente se divide el patrón en dos, obteniendo cuatro regiones, con dos blancas y dos negras intercaladas. De esta manera se consigue depurar la información conseguida en términos de profundidad relativa del píxel con respecto a sus vecinos. Es posible continuar esta división binaria hasta el máximo permitido por el proyector. Por ejemplo, un proyector con resolución 1024x768 necesita $log_2(1024) = 10$ subdivisiones, y por lo tanto para explotar sus características es necesario tomar diez imágenes de la escena con diez patrones diferentes.

- **Códigos grises**
  Con un modo de funcionamiento similar a los códigos binarios, en esta técnica se hace uso de la capacidad del proyector para iluminar utilizando colores. De esta manera, los códigos proyectados se codifican de forma binaria en las señales digitales que son interpretadas como colores.
  Es posible encontrar un código sencillo de escala de grises en Wikipedia[4].

- **Tres cambios de fase (Three phase shift)**
  La técnica *Three Phase Shift*[12] se basa en el modelo matemático explicado en el capítulo 3 que permite gracias a tres imágenes capturadas, cada una de ellas iluminada con un patrón en escala de grises senoidal, evaluar la posición relativa del píxel.
  El método *Three Phase Shift* obtiene nubes de puntos muy densas, esto quiere decir

que calcula un valor de profundidad para cada píxel gracias al uso de un hardware sencillo: una cámara y un proyector.

- **Dos cambios de fase más uno**
  Se trata de una versión del método anterior[22] que mejora la adquisición del color de la escena ya que hace uso de una fase blanca, mediante la cual no es necesario reconstruir el color posteriormente como se explica en la sección 2.3.2.

- **Técnicas experimentales**
  Los proyectores de tipo DLP tienen cierto parpadeo característico que, en conjunción con cámaras de alta velocidad, pueden ser utilizados para recoger la información tridimensional de la escena como se explica en el artículo de Narasimhan Et al.[16].

En Noviembre de 2010, Microsoft pone a la venta Microsoft Kinect 1.4, un dispositivo comercial de luz estructurada para uso con videojuegos. Inmediatamente generó un gran interés en la comunidad de desarrolladores y aparecieron herramientas no oficiales para su uso general[7]. Finalmente, visto el interés de la comunidad, Microsoft liberó una plataforma de desarrollo[8] en Junio de 2011.

Figura 1.4: Sensor comercial de luz estructurada: Microsoft Kinect

De todas las técnicas aquí introducidas, este proyecto se centra en el estudio de la técnica *three phase shift*, ya que es la que ofrece un mejor resultado tanto en tiempo como en calidad haciendo uso de dispositivos domésticos. Es una técnica con unas características muy interesantes desde el punto de vista de la computación y con buenas perspectivas de crecimiento en el futuro.

1.2. Objetivos

- Estudiar y entender *three phase shift*: características de éste, su naturaleza y modo de funcionamiento del mismo.
- Estudiar la relación de los datos de entrada sobre el resultado.
- La etapa *Phase Wrap*: implicaciones en la calidad de la salida.
- La etapa *Phase unwrap*: funcionamiento, coste y complejidad de la etapa.
- Encontrar la manera de facilitar el uso de la técnica para su aplicación con infraestructuras baratas o domésticas.

1.3. Organización de este trabajo

La memoria de este proyecto de investigación está organizada de la siguiente manera: El capítulo 2 es un desarrollo de la técnica *structured light: three phase shift*, analizando sus etapas y evaluando los problemas encontrados.

Los capítulos 3 y 4 tratan sobre las etapas principales, su naturaleza y cómo solucionan el problema para el cual están diseñadas.

En el capítulo 5 se analizan los resultados obtenidos y se intenta dar una explicación a la problemática encontrada.

Por último en el capítulo 6 se analizan las posibles líneas de trabajo a las que este proyecto puede dar paso.

Capítulo 2

Tres cambios de fase (Three phase shift)

Three Phase shift es un método que permite añadir una dimensión de profundidad a una matriz de píxeles. Se parte de tres imágenes de una misma escena, sobre cada una de las cuales se ha proyectado un patrón diferente. Mediante un proceso de reconstrucción, se unifican estas imágenes y es posible evaluar una posición relativa para cada uno de los píxeles en el espacio.

2.1. Descripción del método

El método *three phase shift* consta de varias etapas en las cuales se depura progresivamente la información de profundidad. Para que esto sea posible es necesario que las tres imágenes originales tengan el mismo encuadre de la escena, ya que serán unificadas en una única imagen con profundidad.
A continuación se ilustra en la figura 2.1 las etapas necesarias en el proceso y sus correspondientes datos. Este proyecto se centra en las áreas relacionadas con la computación, éstas son la etapa *phase wrap* y la etapa *phase unwrap*. Una vez concluido el proceso se obtiene una matriz bidimensional con la profundidad de cada píxel.

- **Captura**
  La captura se realiza haciendo uso de tres patrones, los cuales se proyectan sobre la escena a escanear. Estos patrones tienen forma de franjas senoidales en escala de grises que modulan la intensidad con la que se ilumina la escena. Las franjas pueden estar presentes en cualquier orientación pero ésta debe de ser la misma en los tres patrones. La diferencia entre los tres patrones es que estos se encuentran desfasados entre sí 120°.

- **Wrap**
  Phase Wrap es la operación por la cual se extrae la información de profundidad de las tres imágenes originales. Como se explica más adelante, el resultado de ésta es el mapa

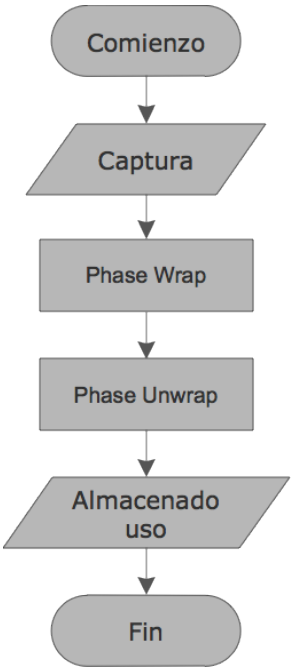


Figura 2.1: Diagrama de flujo del método *three phase shift*

de fase *wrap* en el cual se aprecia la profundidad, pero es necesario refinarla mediante la siguiente etapa. El modelo completo de esta operación se explica en la sección 3.1.

- **Unwrap**
  El mapa *wrap* obtenido en la etapa anterior presenta cierto defecto producido por el cálculo, en el cual existen ciertas discontinuidades que deben ser eliminadas. *Phase unwrap* procesa este mapa para conseguir un mapa de profundidad continuo.

Cada etapa del proceso tiene un juego de datos diferente partiendo de tres imágenes originales. A medida que avanza el proceso se obtienen otros productos:

- **Patrones**
  Se construyen tres patrones en escala de grises que tienen ciertas franjas generadas mediante la modulación de la intensidad con una función senoidal. Cada uno de los patrones es igual al anterior desfasado 120°. En la figura 2.2 se muestra un ejemplo de patrones verticales.

- **Imágenes de fase**
  Estas imágenes corresponden con los patrones anteriormente descritos tras ser proyectados sobre la escena. Al fotografiar la misma se obtiene una imagen con iluminación modulada.

- **Mapa de fase wrap**
  Después de aplicar el algoritmo *phase wrap* explicado en el capítulo 3, se consigue un mapa de fase con discontinuidades al que llamaremos mapa *wrap*.

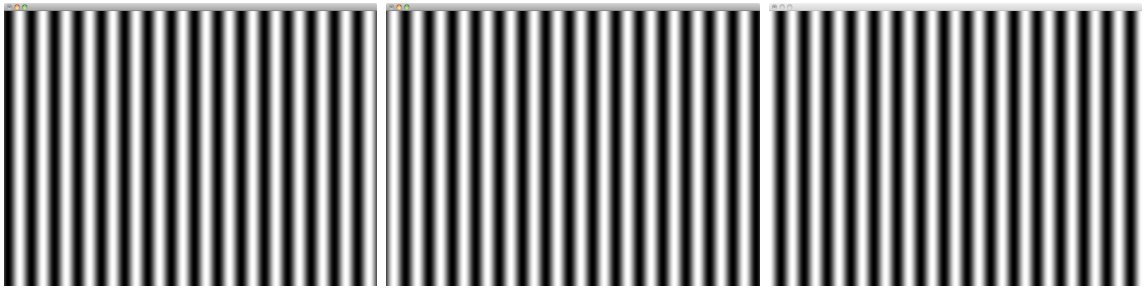


Figura 2.2: Tres patrones proyectados desfasados 120°

- **Mapa de fase unwrap**
  Eliminadas las discontinuidades mediante una de las técnicas *phase unwrap* explicadas en el capítulo 4, finalmente obtendremos el mapa *unwrap* o mapa de profundidad.

Las técnicas utilizadas pertenecen al ámbito de la óptica, haciendo uso de fenómenos físicos que permiten la reconstrucción de un mapa de profundidad en que cada píxel tiene un valor asignado que nos ayudará a reconstruir una nube de puntos que imita la forma geométrica del objeto original. La intención de este trabajo no es modificar o sustituir el modelo matemático, sino estudiarlo desde el punto de vista de la computación, de qué manera atacar este problema con el hardware existente y programar el algoritmo de una forma más eficiente o con resultados más fieles a la realidad. Este planteamiento es contrario a los trabajos tradicionales encontrados, que centran la atención en la infraestructura y que poco es explicado de la problemática algorítmica, como en el artículo de Zhang et al.[21].

A continuación se introducen brevemente las etapas para el método *three phase shift* sobre las que este trabajo se centra.

Phase wrap:

La primera fase del algoritmo es la de unificar las tres imágenes en un único mapa de fase, como se explica en el capítulo 3.
Esta operación se realiza sobre las tres imágenes tomadas y manteniendo el orden de éstas, ya que el orden de la fase es importante. Si las fotografías han sido tomadas a color, será entonces necesario convertirlas a escala de grises antes de proceder, ya que en este caso sólo interesa el valor absoluto de intensidad en el píxel. Las franjas de diferente intensidad serán utilizadas para crear un mapa de fase al que llamaremos mapa *wrap*, el cual tiene unas características muy especiales.
A simple vista, el mapa wrap recuerda al *z-buffer* de una imagen 3D producida por computador[3], una técnica utilizada para almacenar la profundidad de los píxeles en un algoritmo de rasterización. Sin muchas dificultades, somos capaces de percibir la posición relativa de los objetos que aparecen. Pero existe una complicación resultante del método empleado. La operación que evalúa cada píxel hace uso de la operación trigonométrica *arctan* la cual produce una salida con una discontinuidad entre $-\pi/2$ y $\pi/2$.

Phase unwrap:

Llegado este punto, tenemos un único mapa de píxeles con un único valor de fase para cada píxel. El problema es que la anterior etapa nos devuelve una valoración en un intervalo [-π,π] de la fase en el píxel lo cual muestra discontinuidades que son necesarias corregir.

2.2. Aplicaciones

En la literatura se encuentran diversas aplicaciones de *three phase shift*. A continuación se muestran concretamente aplicaciones del algoritmo completo, ya que *phase unwrap* tiene otras aplicaciones distintas, como transformaciones afines realizadas sobre datos acotados entre dos límites que se producen en algunos sensores (como ocurre en un sensor de resonancia magnética[1]):

- **Reconocimiento de gestos faciales**
  Las expresiones faciales tienen gran importancia en nuestra conducta social y son complicadas de sintetizar o analizar. Wang et al.[19] desarrolla un sistema de captura, análisis, catalogación y síntesis para el estudio de la cara humana y sus expresiones.

- **Uso en industria para control de calidad**
  Las técnicas tradicionales de control de calidad en industria se centran en la toma de medidas mediante técnicas de contacto. Éstas proporcionan datos precisos pero son incapaces de modelar todas las características de un objeto, como por ejemplo la profundidad de una perforación. En el trabajo de Bieri y Jacot[9] se muestra cómo es posible aplicar *three phase shift* con ese fin.

- **Conservación del patrimonio cultural**
  El artículo Reznicek y Pavelka[17] se fundamenta en la necesidad de catalogar los daños y el estado de conservación de las esculturas patrimonio de la República Checa. Un escaneado basado en técnicas láser está fuera de alcance debido a su coste económico y de tiempo. Por ello se centran en el uso de técnicas más económicas para cubrir estas necesidades.

2.3. Problemas detectados

En este apartado se enumeran y exponen los problemas detectados, los cuales son el objetivo principal de la investigación propuesta en este trabajo. Estas pruebas han sido realizadas gracias al framework *structured light* ofrecido por el grupo de trabajo structured light mantenido por McDonald y Weber[15]. Durante estas pruebas fueron detectados varios problemas, muchos relacionados con la calidad de la infraestructura de captura, y otros

producidos por el comportamiento natural del algoritmo utilizado en la implementación.

2.3.1. Problemas de captura

A pesar de que en este proyecto no se trata con los problemas naturales de la captura de las imágenes, a continuación se enumeran algunos problemas habituales:

- **Fondo desencuadrado**: en el caso de que exista un fondo, éste debe de mantener la perpendicularidad al plano proyectado. Si no es así, el modelo capturado no mostrará la geometría correcta del fondo. En el caso de que el fondo carezca de interés es posible enmascararlo mediante algún algoritmo de selección y procesar solamente las áreas de interés. De esta manera se obtiene la profundidad de los píxeles pertenecientes al objeto deseado.

- **Saturación del sensor**: el uso del proyector produce que zonas de la imagen puedan estar mucho más iluminadas de lo que convendría, saturando el sensor de la cámara y consiguiendo así imágenes sobreexpuestas. Esto produciría un valor alterado de intensidad para cada píxel, falseando así el color y, lo más importante para este proyecto, la geometría. Los objetos procesados no siempre son de color blanco mate, por lo que la captura de objetos con determinadas características se hace muy difícil o imposible con esta técnica.

- **Deformaciones por la lente**: cualquier deformación en la óptica de la cámara dará como resultado que las líneas no estén correctamente alineadas deformando la nube. Además, hay que tener en cuenta que la cámara tiene una apertura en forma de trapecio, lo cual produce valores de profundidad incorrectos debido a que las distancias relativas entre objetos cercanos serán más acentuadas que las distancias entre objetos lejanos. Esto se debe a que tanto el proyector como la cámara tienen un área de efecto en perspectiva, lo cual no ocurriría con un hipotético sistema ortogonal.

2.3.2. Problemas cromáticos

Existe otro tipo de error producido por la restauración del color de la escena, ya que ninguno de los patrones utilizados realiza una iluminación homogénea de ésta.

Errores al computar el color del píxel, las denominadas franjas oscuras: la reconstrucción del mapa de color se realiza trabajando sobre una interpretación RGB de la imagen. Partiendo de los tres mapas de bits originales, se calcula la media de los tres valores de intensidad para cada una de las tres componentes de cada píxel, de esta manera se consigue una representacion cercana al color de la escena con iluminación homogénea. Sin embargo en la figura 2.3 se muestran los tres valores de fase superpuestos para ilustrar áreas que presentan valor de intensidad pobre.

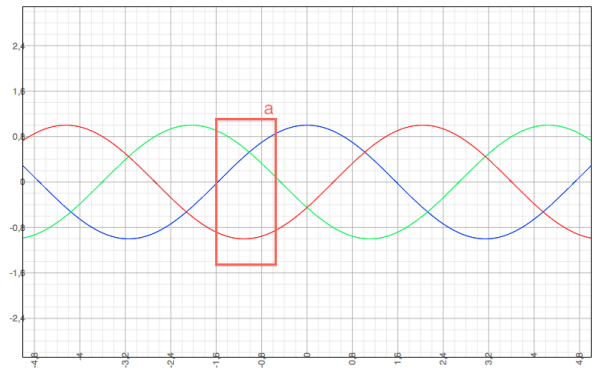


Figura 2.3: Un error al decodificar la imagen produce mala interpretación del color

Dado que se usan tres senoidales con un desfase respectivo de la anterior, al recomponer las componentes de color existe una diferencia de intensidad entre la primera y segunda fase, entre la segunda y tercera y entre la tercera y la primera + 2π. No es posible recomponer una señal de intensidad correcta. Esto produce franjas en las cuales el color sufre cierto "desvanecimiento".
El trabajo de Zhang y Yau[22] intenta solucionar esta característica mediante la sustitución de uno de los patrones por una imagen blanca en la cual todo el área está iluminada. En este proyecto no se ha trabajado mucho sobre el color, centrando los esfuerzos sobre la reconstrucción de la tridimensionalidad.

2.3.3. Problemas algorítmicos

Por último se describe un problema relacionado con la naturaleza del algoritmo:

Áreas con una interpretación de la profundidad errónea: en la figura 2.4 se observa que existen áreas con una profundidad mal interpretada: los detalles (a) y (b) muestran un artificio creado a partir de los dobleces de la camiseta. En la imagen la región de dichos dobleces esta separada del resto de la camiseta mediante sombras, que hacen que el algoritmo sea incapaz de identificar dicha región como parte del mismo objeto. En algunas ocasiones este efecto ocurre sobre regiones originalmente contiguas, dividiendo objetos.

Tras aplicar la técnica *three phase shift* esta región aparece con un valor de profundidad inadecuada, creando discontinuidades donde debería haber segmentos próximos. En algunos casos este efecto puede provocar que regiones habitualmente continuas aparezcan dividas, desfigurando completamente el objeto en cuestión. Los artificios generados deben ser tratados ya que malinterpretan completamente la geometría de la escena. Este problema relacionado con el algoritmo *phase unwrap* es uno de los aspectos más relevantes de esta investigación y depende de la implementación especifica de este algoritmo como se explica en el capítulo 4.

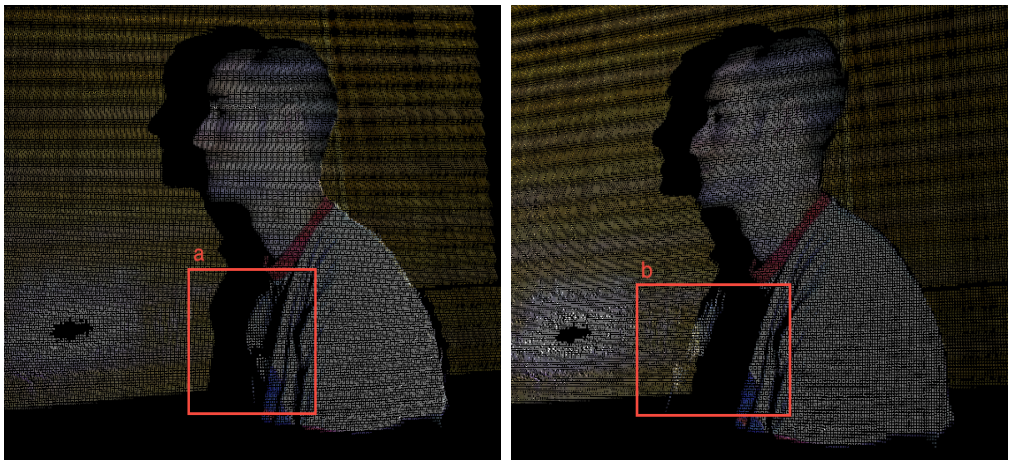


Figura 2.4: Un error al decodificar la imagen produce áreas mal interpretadas

Los errores detectados en este apartado han sido de gran importancia para la evaluación de las áreas donde es necesario realizar más trabajo para la mejora de esta tecnología. Una vez identificados estos errores, se analizan las etapas del algoritmo para buscar sus orígenes y proponer soluciones. Lamentablemente por limitaciones de infraestructura, los problemas de captura no han podido ser resueltos ya que hubiera sido necesario disponer de proyectores y cámaras con mejores prestaciones como se explica en la sección 6.2.

Capítulo 3

Phase wrap

Tras la captura original de las tres imágenes con proyección de patrón comienza el proceso de reconstrucción del mapa de profundidad. La primera etapa de la técnica *three phase shift* consiste en realizar la transformación de las tres imágenes con patrones diferentes en una única imagen llamada mapa *wrap* o mapa de fase, en la cual se obtiene un mapa con información sobre la profundidad relativa del píxel. Este mapa tiene una característica principal: está acotado entre $-\pi$ y π. El mapa *wrap* tendrá tantas discontinuidades de este tipo como franjas tenga nuestro patrón original. En la figura 3.1 se puede ver la entrada y el subsiguiente mapa *wrap*.

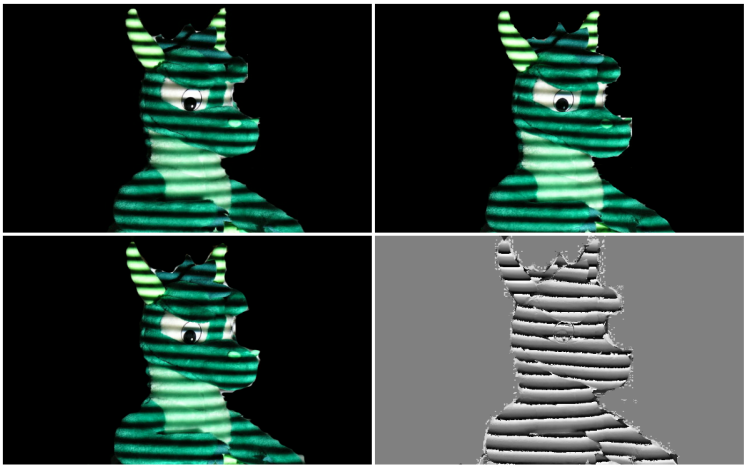


Figura 3.1: Imágenes de entrada wrap y resultado posterior

Phase Wrap en el cambio de fase es el proceso de determinar los valores de fase de las franjas del patrón dentro de un rango [-π;π] [-0,5;0,5] ó [0;1] dependiendo de cómo se implemente la operación. De esa manera obtenemos la modulación en fase de la intensidad del píxel, la cual es un indicativo de la profundidad a la que se encuentra el objeto representado en el píxel dado. Posteriormente utilizaremos el valor de fase calculado como indicador de la profundidad relativa del píxel[13].

A continuación se explican los detalles relativos al procedimiento y los requisitos previos necesarios.

3.1. Operación de *phase wrap*

Una explicación completa del modelo puede ser encontrada en el trabajo de Wang[20]. A continuación se resumen las ecuaciones que hacen posible la reconstrucción de la información de profundidad. Partiendo de tres imágenes con patrones diferentes proyectados sobre la escena, cada uno desfasado de la anterior, estas imágenes tendrán la forma de las ecuaciones (3.1), (3.2) y (3.3).

$$I_1(x,y) = I'(x,y) + I''(x,y)cos[\phi(x,y) - \alpha] \quad (3.1)$$
$$I_2(x,y) = I'(x,y) + I''(x,y)cos[\phi(x,y)] \quad (3.2)$$
$$I_3(x,y) = I'(x,y) + I''(x,y)cos[\phi(x,y) + \alpha] \quad (3.3)$$

I'(x,y) es la intensidad media correspondiente a la luz ambiente. I'' es la intensidad modulada, la cual es proporcionada por el patrón proyectado. ϕ(x,y) es la fase con la que se trabaja y α el paso del cambio de fase. La luz ambiente corresponde a la iluminación de la escena sin proyectar patrón alguno, sobre esta iluminación será añadida la iluminación propia del patrón, de forma senoidal (coseno en este caso).

Resolviendo el sistema, podemos recuperar la señal ϕ(x,y) como se ilustra en la ecuación (3.4).

$$\phi(x,y) = \arctan(\sqrt{3}\frac{I_1 - I_3}{2I_2 - I_1 - I_3}) \quad (3.4)$$

3.1.1. Pre-requisitos

En el caso de que la captura sea a color es necesario obtener tres imágenes en escala de grises. Para ello, si las imágenes son en color es necesario convertirlas a un único valor de intensidad. Como se trata de un valor de intensidad para el cálculo de la fase no es necesario realizar ninguna operación sofisticada, como ponderar las componentes en función de la sensibilidad del ojo humano. Con la media aritmética basta. Sin embargo las implementaciones con las que se ha trabajado[15] realizan esta media ponderada aunque no se ha encontrando diferencias significativas en los resultados.

3.2. Análisis de phase wrap

En la siguiente sección se analizan dos aspectos concernientes a la etapa *phase wrap*, primero se realiza un estudio sobre diversas técnicas para acelerar la operación y a continuación se analizan técnicas para la corrección de errores y la mejora de la calidad.

3.2.1. Aceleradores

Para acelerar la operación existen técnicas que directa o indirectamente aceleran el cálculo. La primera de las técnicas aquí explicada no es una técnica de aceleración en sí misma pero consigue reducir el volumen de datos a procesar y repercute en el rendimiento.

- **Enmascarado**. La operación de enmascarado elimina los píxeles de peor calidad o con resultados erróneos (por ejemplo, división por cero). Si no superan cierto umbral de calidad serán marcados como no válidos y excluidos en la etapa *phase unwrap* y posteriores. Se les asignará un valor de profundidad infinita o nula. A partir de este momento esos píxeles serán ignorados, lo que puede producir una pérdida de información profundidad por los ruidos o por la bondad del umbral.

- **Lookup Table**[6]. La operación *arctan* no es sencilla y resulta costosa en términos de tiempo. Por ello podemos definir una tabla con los resultados de la operación de forma que posteriormente se acceda a estos para evitar el cálculo. Lookup table es una estructura de datos que permite almacenar los resultados previamente calculados de la operación y después utilizarlos gracias a un sistema de indexación.
  En este caso la operación que corresponde a la ecuación 3.4 será almacenada mediante dos índices, siendo uno de ellos el numerador de la operación y el otro el denominador. Los resultados de la multiplicación por $\sqrt{3}$ y la posterior operación *arctan* se encuentran pre-calculados en una tabla de dos dimensiones. Por otra parte el indexado se hace discretizando los dos espectros, tanto el denominador como el numerador, por lo que dependiendo del tamaño en memoria que utilicemos reduciremos así mismo la precisión del cálculo.

- **GPGPU (*general purpose GPU*)**. Dada la naturaleza del problema es sencillo realizar la operación masivamente en paralelo en una o varias tarjetas de vídeo GPU. Todas las operaciones necesarias para computar un píxel dependen exclusivamente del valor de esa misma posición en las tres imágenes originales y el resultado se escribe una única vez en la posición final, por lo que es perfectamente paralelizable. Un código sencillo para implementar esta operación en CUDA se muestra en la figura 3.2.

```
__global__ void phaseWrap (float *out,
                           unsigned char *phase1,
                           unsigned char *phase2,
                           unsigned char *phase3,
                           unsigned i){

    // para cada elemento de entrada
    unsigned thid = threadIdx.x + BlockIdx.x * BlockDim.x;
    if (thid < i){
        // computa denominador y numerador
        float denom = (2*(phase2[thid]) - phase1[thid] - phase3[thid]);
        float numer = (phase1[thid]-phase3[thid]);
        // computa valor y salva resultado
        out[thid] = (atan2(sqrt(3.0)*numer, denom));
    }
}

int main (){
   // establece el entorno y copia la entrada a GPU
   ...

   // los parámetros de la imagen en escala de grises
   unsigned w = imagen.Width;
   unsigned h = imagen.Height;

   // realiza la llamada
   dim3 block = 256;
   dim3 grid = ceil((h*w)/256);
   phaseWrap<<<grid, block>>> (salida, phase1, phase2, phase3, w*h);

   // recupera la entrada de GPU
   ...
}
```

Figura 3.2: Ejemplo de una posible implementación en GPU usando CUDA

3.2.2. Breve estudio sobre los filtros espaciales

El mapa *wrap* tiene unas características particulares que deben ser mantenidas para conseguir posteriormente el mapa de profundidad. Tras realizar la operación *wrap* se obtiene un mapa con magnitudes acotadas entre $-\pi$ y π. Cualquier ruido que aparezca en una de las imágenes originales es transmitido a este mapa y, posteriormente, al mapa de profundidad. En la figura 3.3 se pueden ver las siguientes características correspondientes a una cara humana: se puede percibir cierta variación de la fase en el corte vertical en la zona que sería la ceja (a) y también podemos observar las discontinuidades (b) producidas por la operación *arctan*. Estas discontinuidades aparecen en forma de asíntotas verticales, sin embargo aparecen ciertos problemas: en el corte horizontal se puede observar que en la región de la mejilla existe cierto ruido (c), producido probablemente por la pobre respuesta del sensor a los cambios de intensidad.

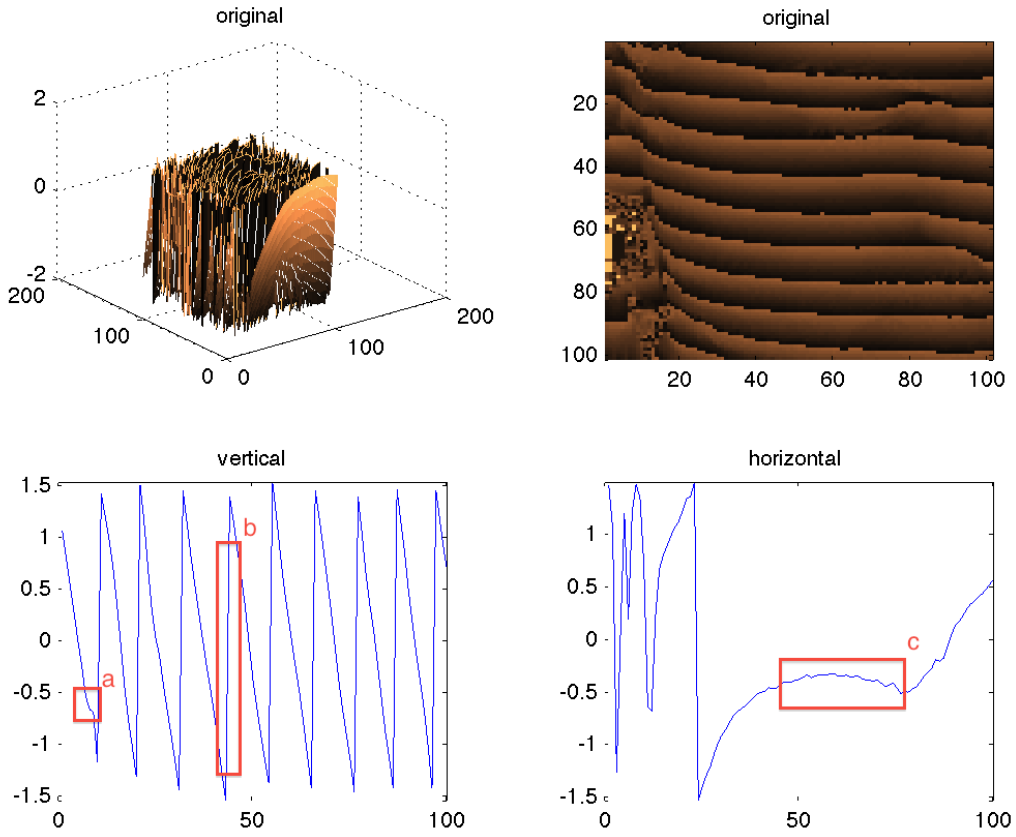


Figura 3.3: Mapa *wrap* original sin aplicar filtro alguno

Como se ha explicado anteriormente el mapa wrap se encuentra acotado entre $-\pi$ y π, estas discontinuidades se eliminan en la etapa *phase unwrap* explicada en el capítulo 4. Sin embargo existen errores que pueden ser confundidos con estas discontinuidades naturales.

La naturaleza del error que se desea evitar es de alta frecuencia, son las discontinuidades cercanas a 2π no producidas por *arctan* que pueden provocar una interpretación equivocada de la profundidad. Estos ruidos deberían ser eliminados ya que al aplicar transformaciones afines se transmiten a los vecinos y propagan el error. Es por este motivo que en esta sección este proyecto se ha centrado en analizar algunos filtros para realizar correcciones y de esta forma estudiar qué impacto tiene el uso de diferentes técnicas.

Media

El filtro de la media obedece la ecuación (3.5).

$$p(x) = \frac{\sum_{i=-n}^{n} \sum_{j=-n}^{n} p(i,j)}{(2 * n + 1)^2} \tag{3.5}$$

Este filtro, ya sea de forma circular o cuadrada, no aporta mucho al problema tratado. Su comportamiento homogeneiza los valores y elimina las discontinuidades. En la etapa del *phase unwrap* se van a eliminar estas discontinuidades haciendo una curva continua. Al reducir la pendiente de estas discontinuidades la etapa posterior será incapaz de detectarlas y por tanto corregirlas.
Este tipo de filtros consiguen reducir el ruido puntual pero confunden los bordes y son fatales para el progreso del algoritmo.

Gausiano

El filtro Gausiano no deja de ser un filtro de la media en la cual se ponderan los píxeles haciendo uso de la distribución normal. Al igual que la media produce un efecto de suavizado sobre las áreas de alta frecuencia por lo que afecta también a las discontinuidades y de esta manera no resulta útil en este caso.

Motion

El filtro de las herramientas para tratamiento de imágenes de Matlab[2] nos devuelve la aproximación de un movimiento lineal de la cámara, desplazando en cierto ángulo una determinada distancia expresada en píxeles. La figura 3.4 ilustra el resultado de la operación tras convolucionar.

Si nos fijamos en las características marcadas resaltadas en la figura 3.4 en la imagen derecha correspondiente al corte horizontal podemos ver cómo se consigue una superficie muy suave en la zona de la mejilla (c) y en la misma imagen, más a la derecha (d) la nariz. En el detalle vertical también podemos identificar la ceja (a). Sin embargo las discontinuidades propias del mapa de fase *wrap* se encuentran distorsionadas (b, e) ya que presentan una pendiente menor y, por lo tanto, la siguiente etapa será incapaz de recuperar la imagen continua sin complicar la lógica de ésta.

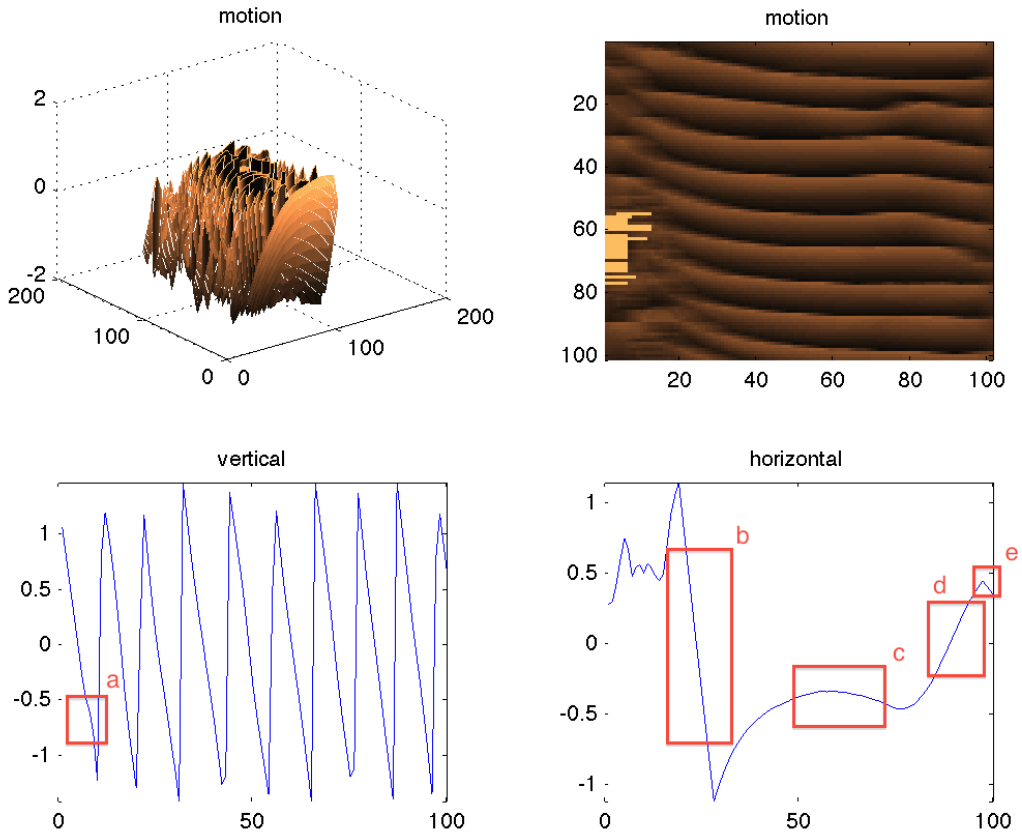


Figura 3.4: Características del mapa *wrap* después de aplicar un filtro motion

Laplaciano

El filtro, como se explica en el libro de Pajares Martinsanz y Cruz García (Visión por Computador[14] 150-151) es un operador basado en la segunda derivada, por lo que resulta bueno para la detección de bordes y cambios de alta frecuencia. Los coeficientes asociados con el píxel central y los coeficientes asociados al resto de los píxeles son negativos y la suma de las componentes del operador Laplaciano es cero, por lo que en regiones de baja frecuencia o intensidad uniforme el resultado será próximo a cero. Todo lo contrario ocurre en zonas de alta frecuencia donde exista un borde.
Como se puede ver en la figura 3.5 tras aplicar el filtro se conservan bien la verticalidad de las discontinuidades (a). Sin embargo se pierde la mayor parte de la información sobre los relieves (b), lo cual lo hace inviable para su uso como corrector. Por otra parte sería posible el uso de este filtro para guiar correcciones más agresivas sobre áreas de interés.

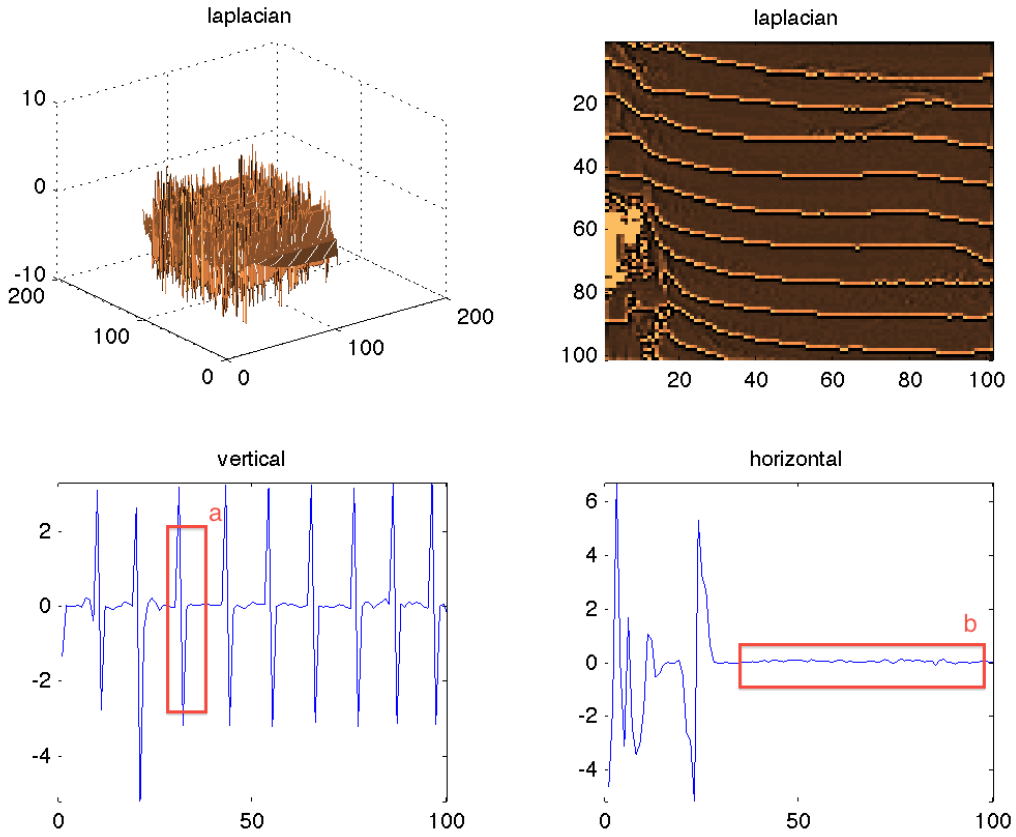


Figura 3.5: Características del mapa *wrap* después de aplicar un filtro laplaciano

Enfatizado de bordes

Los operadores de borde son casos particulares de operadores de la primera derivada para enfatizar bordes en un sentido determinado (Visión por Computador[14] 134-149), ya sean bordes verticales u horizontales.
Rompe absolutamente con la posibilidad de recuperar la información de profundidad. Este filtro destruye las curvas significativas y resalta los bordes, los cuales deben ser eliminados en una etapa posterior, por lo que carece de sentido para el problema a tratar.

3.2.3. Filtro Escogido

Los filtros utilizados para la reconstrucción de imágenes en el dominio espacial se basan en que cada píxel depende del valor de los vecinos y tienen un aporte en función de la distancia entre ellos. Esto se debe a que, intuitivamente, si existe ruido sobre un píxel el color original del píxel debe ser similar a los vecinos. Es precisamente esta situación la que

se quiere evitar en el algoritmo de cambio de fase.
El filtro bilateral[18] está diseñado para suavizar las imágenes mientras que conserva los bordes. Debería conseguir buenos resultados ya que preserva los límites de nuestro mapa *wrap*, y no debería alterar el comportamiento de la siguiente fase del algoritmo.

Como se puede ver en la fórmula (3.6) se trata de un filtro que permite pasar las bajas frecuencias, el cual elimina muy bien los ruidos de tipo puntual sin embargo no respeta discontinuidades.

$$h(x) = k_d^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) d\xi \tag{3.6}$$

La función c(ξ,x) mide la cercanía del punto ξ al píxel evaluado x, como el sumatorio de estas componentes depende de la distancia y el tamaño del filtro, es necesario añadir una constante de normalización k_d.
La formula (3.7) corresponde a un filtro de rango en el cual la función s evalúa la similitud del valor del píxel en la posición ξ con respecto al píxel en x.

$$h(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(f(\xi), f(x)) d\xi \tag{3.7}$$

La combinación de ambos filtros (3.8) corresponde finalmente al filtro bilateral. Se puede observar que la imagen resultante depende ahora de la distancia entre el píxel en ξ y en x pero también depende de los valores de estos píxeles.

$$h(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi \tag{3.8}$$

En teoría este filtro resulta ideal para aplicar al problema. Es necesario eliminar toda aquella discontinuidad próxima a 2π que no sea producida por *phase wrap*, de esta manera sólo deberían ser corregidas aquellas discontinuidades que no alteren el mapa de profundidad. En la sección 5.1.1 se muestra un estudio sobre los resultados conseguidos con la utilización de este filtro.

Capítulo 4

Phase Unwrap

Tras la operación de la etapa *wrap* es necesario eliminar las discontinuidades producidas por la operación arco tangente (*arctan*). Para ello se realizará una reconstrucción del mapa de profundidad en el cual se trasladarán en magnitud los valores dependiendo de las discontinuidades próximas. La salida de *arctan* está acotada entre -π y π, o lo que es lo mismo: una salida módulo 2π. En los datos empíricos encontraremos unas discontinuidades próximas a 2π. *Phase unwrap* o "desenrollado" de fase, es la operación que elimina estas discontinuidades convirtiendo la salida en una función continua.

La operación *phase unwrap* ha sido objeto de investigación debido a sus diversas aplicaciones en diferentes campos. Es considerada una operación computacionalmete costosa cuyo resultado es susceptible al orden de actuación, por lo cual un resultado correcto no está siempre garantizado.

4.1. Operación básica

Como puede verse en la figura 4.1 la idea detrás de *phase unwrap* no es más que eliminar discontinuidades producidas por una señal de entrada acotada entre dos valores.

En la figura 4.2 se muestra el código C de la operación básica *unwrap* en la cual el valor de fase de un píxel es corregido con cierta desviación si la diferencia con su vecino sobrepasa cierto umbral.

En este caso son los datos que se utilizan en los experimentos donde las discontinuidades corregidas están acotadas dentro del intervalo [-0,5 0,5] el cual se ha utilizado para mayor comodidad en lugar de la típica discontinuidad de 2π. La operación se lleva a cabo de la siguiente manera: se evalúa la diferencia entre dos valores adyacentes, si son superiores a cierto umbral el salto entre ellos se corrige. Dicho umbral se define en función del intervalo en uso, por ejemplo: en el caso del intervalo antes citado si el valor absoluto de la diferencia entre dos píxeles excede 0,5 entonces debe ser corregido.

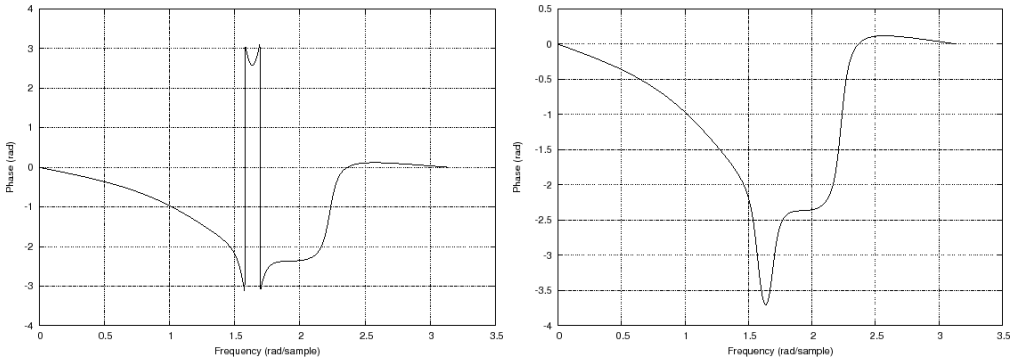


Figura 4.1: Operación de *phase unwrap* sobre discontinuidades 2π

```
float a = wrapped[i];
float b = source;
float diff = a - (b - (int) b);
if(diff > .5)
    diff--;
else if(diff < -.5)
    diff++;
unwrapped[i] = b + diff;
```

Figura 4.2: Código C de la operación básica de unwrap

Existen dos correcciones posibles las cuales se corresponden con las dos ramas de la sentencia condicional. En la figura 4.3 se ilustran las dos transiciones posibles: negativas (a) donde los valores pasan del límite superior al inferior produciéndose un salto de -2π y positivas (b) con un salto de $+2\pi$. En el caso de la figura mostrada los límites se encuentran entre 0 y 255 ya que son los posibles valores en una imagen de escala de grises de 8 bits, sin embargo esto es así para poder mostrar el resultado de la operación en forma de imagen mientras que internamente se utiliza *float* que permite expresar valores entre 10^{37} y -10^{37}.

4.2. Transformación unwrap

Existen diferentes formas de realizar la operación *phase unwrap*. Las catalogaremos dependiendo de la tecnología de programación utilizada. A medida que avanza la lista se nombran métodos más sofisticados y rápidos pero no en todos los casos con mejores resultados.
Como catalogan Arevallilo Herráez et al. en su trabajo[11] existen tres familias:

1. **Algoritmos Globales**: formulan el problema en términos de minimización, el cual es el error producido en las discontinuidades. Constan de sucesivas iteraciones en busca

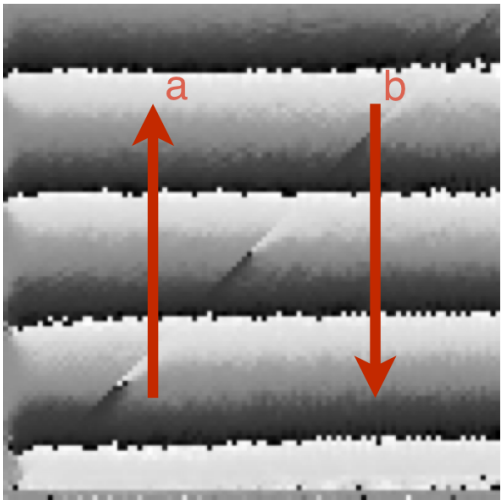


Figura 4.3: Direccion de unwrap

de un mínimo de error por lo que resultan costosos computacionalmente pero por otra parte son los más robustos frente a ruidos.

2. **Algoritmos por regiones**: la imagen es dividida en áreas o regiones que son procesadas por separado y después se realiza un ajuste de cada una de las anteriores regiones con respecto a las regiones vecinas ya computadas. Esta aproximación tiene una buena relación de coste frente a robustez y existen dos categorías:

   - ***Tile-based* (basado en celdas)**: este método evalúa regiones separadas arbitrariamente en función de la posición del píxel.
   - **Region-based (basado en regiones)**: realiza una aproximación más sofisticada de las regiones de forma que los píxeles en la región tengan características similares.

3. **Algoritmos de recorrido**: el recorrido intenta buscar una solución lineal al problema haciendo una única pasada e intentando corregir la fase del píxel en función de los vecinos. Para los algoritmos de recorrido existen también tres categorías:

   - **De camino**: detectan los bordes o los cambios de fase abruptos y utilizan esta información para calcular los desplazamientos de fase a aplicar. Dentro de esta categoría existen métodos en espiral o en dirección múltiple. Son rápidos (carecen de capacidad de vuelta atrás) pero muy vulnerables al ruido dado que el camino a evaluar es inmutable.
   - **Compensación de residuo**: buscan residuos y colocan cortes entre los residuos positivos y negativos. Son algoritmos eficientes también pero poco robustos.
   - **Guiados por calidad**: dada una función de calidad que se aplica a los valores de fase calculados en *phase wrap* es posible evaluar los píxeles que tienen un mejor

valor para ser procesados antes y demorar los que presentan valores fiables. De esta forma se consigue un algoritmo computacionalmente eficiente y con buenos resultados. Un estudio sobre el comportamiento de estos se realiza a continuación en el apartado 4.3.1.

Cada categoría de *phase unwrap* puede ser implementada de diferentes maneras, como se muestra a continuación existen diferentes algoritmos con resultados más o menos afortunados. En este proyecto nos hemos centrado en los algoritmos de recorrido ya que solucionan el problema en orden lineal con buenos resultados.

4.3. Transformaciones en recorrido

En la siguiente sección se describen los métodos estudiados basados en recorrido para la realización de *phase unwrap*. La complejidad de estos algoritmos es lineal, necesitando una única pasada por píxel para corregir su valor. La diferencia entre diferentes transformaciones en recorrido es la forma de escoger el camino mediante el cual se resuelve el mapa: las diferentes elecciones resultan en diferentes interpretaciones. El interés principal de esta sección es identificar la implicación del camino sobre determinadas características y estudiar la naturaleza del problema de la etapa *phase unwrap*.

En la figura 4.4 se muestra a la izquierda el mapa *wrap* y a la derecha una de las tres imágenes originales de uno de los experimentos, en ésta se puede observar que se trata de uno de los bustos escaneados. Para el ojo humano ya es posible percibir la profundidad de la imagen en el mapa wrap, ya que al tratarse de un objeto reconocible el aparato cognitivo humano induce la profundidad del objeto basado en nuestro conocimiento. Sin embargo para la máquina esto se convierte en una operación costosa y absolutamente independiente del contexto, esto quiere decir que no se basa en un sistema de conocimiento como el cerebro.

Algoritmo Top-Down

Se ha realizado una implementación sencilla para analizar la naturaleza del problema a la cual se ha llamado algoritmo Top-Down ya que hace un recorrido independiente por columnas de arriba hacia abajo. Esta implementación es altamente susceptible al error en el mapa de fase tras la operación *phase wrap*. El método Top-Down evalúa cada columna de la imagen de manera independiente de forma que no se tienen en cuenta los vecinos adyacentes en la horizontal. Este comportamiento produce que los píxeles con ruido propaguen su error a los sucesivos elementos de la misma columna e implica que el resultado sea un mapa con diferencias muy grandes entre columnas adyacentes.

Como se explica en el apéndice A.1 las columnas son tratadas individualmente y el valor del píxel se computa dependiendo del valor del anterior. Esto último es común a todas las soluciones basadas en recorrido: los píxeles son corregidos en función de uno anterior

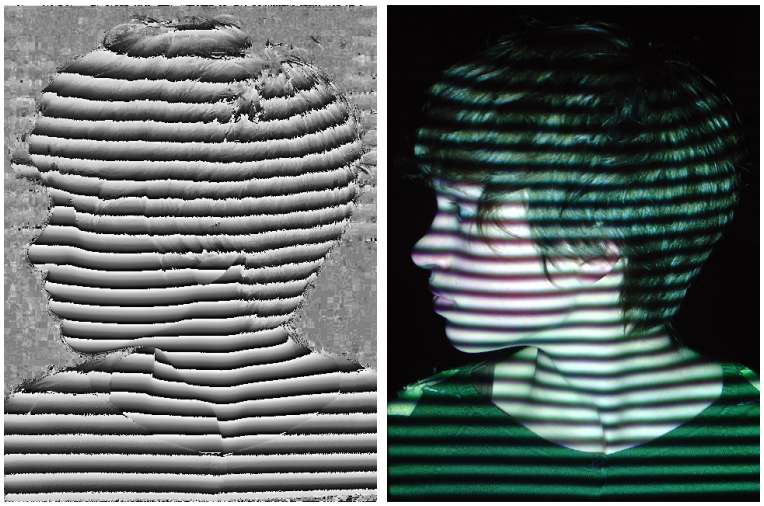


Figura 4.4: Mapa *wrap* utilizado en los experimentos

ya corregido. Pero en este caso en particular al tratarse cada columna como un problema aislado se agrava el efecto del ruido o de las discontinuidades propias de la escena, este comportamiento se puede apreciar en la figura 4.5.

Se puede distinguir la forma del objeto en la imagen pero inmediatamente se aprecian valores erróneos. La región de la barbilla (a) muestra como se ha evaluado en cada columna la discontinuidad, aún así no es correcto y produce un valor correspondiente a una discontinuidad dentada. En la zona del pelo (b) un salto producido por un área de alto contraste se convierte en un indicador de profundidad relevante alterando el comportamiento de toda la columna.

Algoritmo de inundación

El algoritmo de inundación[23] es el más extendido. Se basa en la idea de que dos píxeles vecinos deben tener un valor de fase similar en el mapa *wrap*. Por ello cada píxel evaluado encola a sus cuatro vecinos para ser procesados utilizando una cola FIFO. De esta manera, los cuatro vecinos: norte, sur, este y oeste serán procesados antes de proceder con los vecinos sucesivos. El orden con el que se encolan éstos es independiente del algoritmo y de este orden depende que se muestren determinados errores, los cuales son ilustrados en la figura 4.6.

Se puede observar que se define muy bien el borde de la camiseta, los músculos del cuello (c) o el párpado. Sin embargo aparecen errores acumulados en forma de líneas horizontales (b) que nos recuerdan a los errores identificados en la sección 4.3 para el método Top-down. Estos errores se producen sin duda por la prioridad implícita que se le aplica a los vecinos. También en la zona conflictiva del pelo (a) se produce este error horizontal. Al encolar primero los vecinos en la horizontal se produce el efecto de propagación de error en la horizontal, lo cual no puede ser evitado alterando este orden: alterar el orden produciría que el error se manifestara de forma vertical o en zigzag.

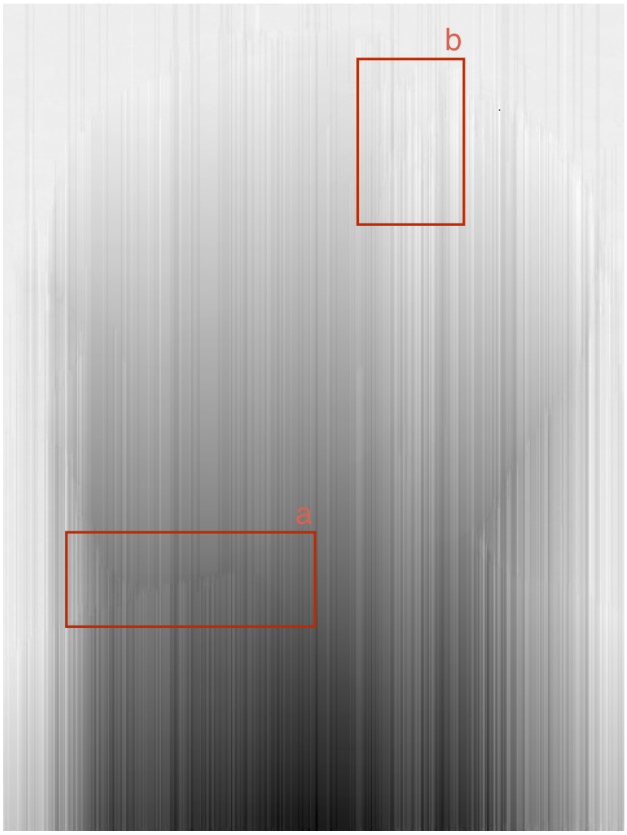


Figura 4.5: Resultado tras realizar *phase unwrap* top-down

Esto nos lleva a pensar que, al igual que el comportamiento del filtro bilateral, la transformación de fase no debe realizarse basada exclusivamente en la proximidad del vecino, sino que también es necesario tener en cuenta la proximidad en fase. Esto nos lleva a guiar la evolución del algoritmo mediante una evaluación de la calidad de cada píxel.

4.3.1. Evaluación de la calidad

Para conseguir mejores resultados es posible guiar el recorrido para evaluar primero regiones más interesantes. Determinados objetos pueden ser continuos en la escena mientras que sufren discontinuidades en el mapa de profundidad final.
El la figura 4.7, realizada con el framework *structurted light*, se puede observar que ambos brazos sufren un corte y que las manos tienen otro valor de profundidad (detalles a y b).

Esto se debe a que el algoritmo (de inundación en este caso) procede verticalmente hacia las manos y horizontalmente hacia el hombro. El orden fijo de ejecución produce que las manos sean evaluadas con anterioridad al codo. La evaluación de este objeto se realiza en dos momentos diferentes y con dos referencias diferentes.
La solución a este problema puede ser realizar una selección más sofisticada del camino, el cual evaluará los píxeles vecinos y de características similares sucesivamente dejando la

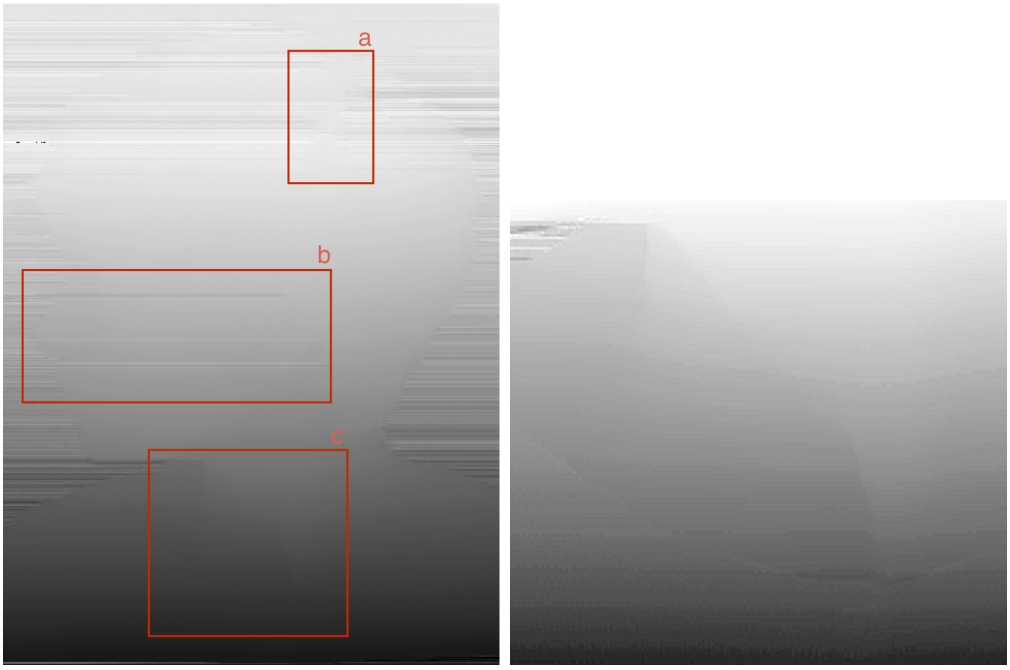


Figura 4.6: Resultado tras realizar *phase unwrap* por inundación y ampliación

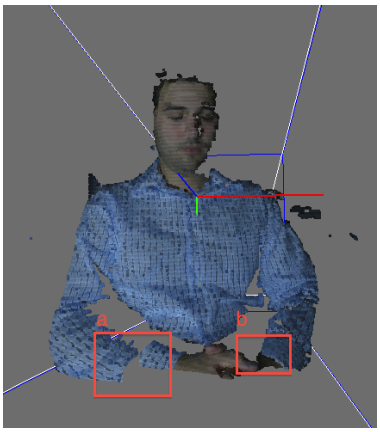


Figura 4.7: Discontinuidades producidas por el recorrido a lo largo de la imagen

decisión de corregir una discontinuidad para más adelante. En la figura 4.8 se puede observar en la imagen de la izquierda: al tener en cuenta solamente la vecindad, el algoritmo "saltará" la discontinuidad entre las manos y el torso de forma que evaluará la discontinuidad antes de evaluar todo el brazo. Si por el contrario se evalúan las discontinuidades del brazo siguiendo la forma, se consigue que el objeto entero tenga un mapa continuo de profundidad, como sugiere el detalle de la derecha.

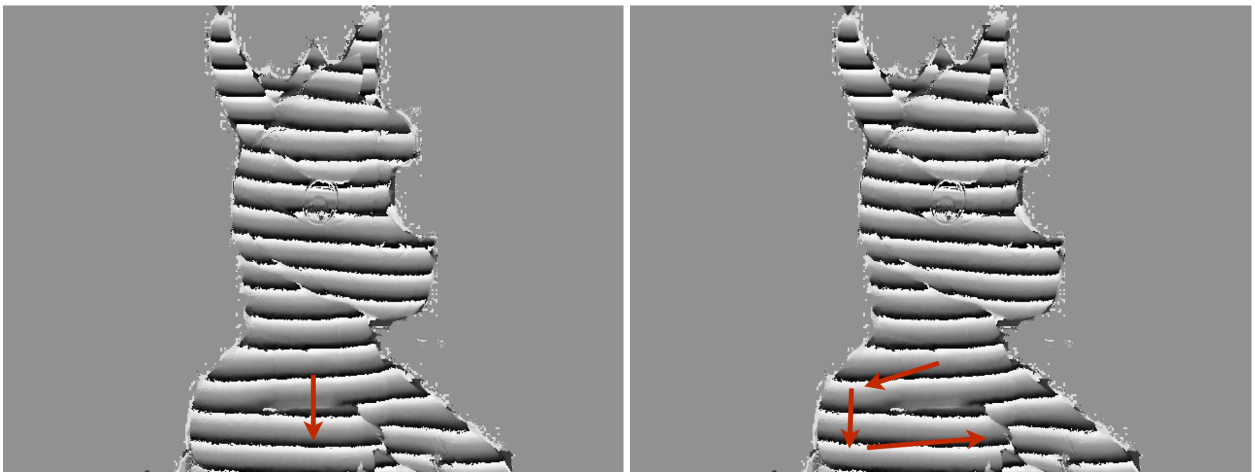


Figura 4.8: Cambio de recorrido basado en la proximidad

Características de calidad

Para priorizar el camino de *unwrap* existen dos técnicas documentadas:

- **Valor del píxel**: se le asigna un valor de calidad a cada píxel dependiendo de la similitud con sus vecinos y dependiendo también de la calidad de la captura original. En la figura 4.9 se ilustra el código C para calcular la calidad de cada píxel. La calidad de cada píxel almacena la suma de las diferencias con sus vecinos, posteriormente es dividida por el rango dentro del cual se encuentran los tres valores de intensidad para el píxel en las tres imágenes originales. Este cómputo es la prioridad con la que este píxel será evaluado, cuanto menor sea el número antes será procesado.

```
float a    = wrap[i];
float up   = wrap[i-weight]
float down = wrap[i+weight];
float left = wrap[i-1];
float right= wrap[i+1];
float distancia =   diff(a, up) +
                    diff(a,down) +
                    diff (a, left)+
                    diff(a, right);
float rango = getRange(phase1[i], phase2[i], phase3[i]);
calidad[i] = distancia / rango;
```

Figura 4.9: Cómputo de la calidad del píxel

Como se puede observar en el código 4.9 al dividir por el rango, los píxeles cuyos tres píxeles padres (los píxeles en las tres imágenes originales capturadas con patrón)

cubran poco rango tendrán mayor puntuación que aquellos cuyos originales cubran un rango mayor y por lo tanto serán procesados después.

- **Valor de la transición**: según el trabajo de Arevallilo Herráez et al.[11]. puede resultar más ventajoso evaluar la calidad en función de la transición a seguir y no como un valor de calidad absoluto para el píxel. Esto quiere decir que el camino a seguir para realizar el "desenrollado" se escoge en función de qué píxel vecino es mejor para continuar respecto a los ya evaluados, y supone un cambio sutil de modo de funcionamiento respecto al anterior algoritmo, que evoluciona de manera que el píxel mejor puntuado se corrige a continuación. Sin embargo éste puede estar bien puntuado gracias a su similitud con vecinos no evaluados todavía, mientras que pueden existir candidatos por evaluar que guardan coherencia con la región evaluada, píxeles similares a la región ya procesada deben ser procesados antes, independientemente de si globalmente su calidad es menor que la de otros.

Una vez analizadas las opciones para priorizar estudiaremos los resultados obtenidos: para ello se ha implementado el algoritmo de inundación substituyendo la cola por una cola de prioridad. Dicha cola ordena por calidad los píxeles para procesar primero aquellos que tengan mejor puntuación.

4.3.2. Algoritmo de inundación guiado por prioridad

Se basa en la idea de que dos píxeles con valor de fase similar en el mapa *wrap* deben tener un valor de fase similar en el mapa *unwrap*.

Para ello se procede de la misma manera que en el algoritmo de inundación con la diferencia de que en este caso, tras encolar los píxeles vecinos, comenzaremos a desencolar por los que muestren un mayor valor de calidad. De esta forma los píxeles con fase similar serán "*desenrollados*" con respecto a los vecinos más próximos en fase.

La evaluación de la calidad del píxel se puede realizar de tres maneras: con anterioridad al proceso en un proceso independiente, simultáneamente con *phase wrap* o a medida que se produce la etapa *phase unwrap*. Ejecutarlo junto a *phase wrap* es el método más rápido ya que hace uso de los mismos datos consumidos o producidos en éste y se beneficiaría de la ayuda de la cache del sistema. Por otra parte, realizar el cálculo mientras se produce la corrección de las discontinuidades puede favorecer la evaluación de calidad, haciendo ésta más precisa, sin embargo es necesario acceder a datos que, de forma normal, no serían necesarios en esta etapa (como, por ejemplo, las tres imágenes originales).

La figura 4.10 muestra como el algoritmo basado en prioridad soluciona algunos de los errores anteriormente comentados. De izquierda a derecha y arriba hacia abajo se ilustra: una de las tres fases originales (a), error propagado verticalmente por el algoritmo top-down

(b), el mismo error se propaga horizontalmente en el algoritmo de inundación (c) y finalmente el algoritmo basado en prioridad del píxel(d).

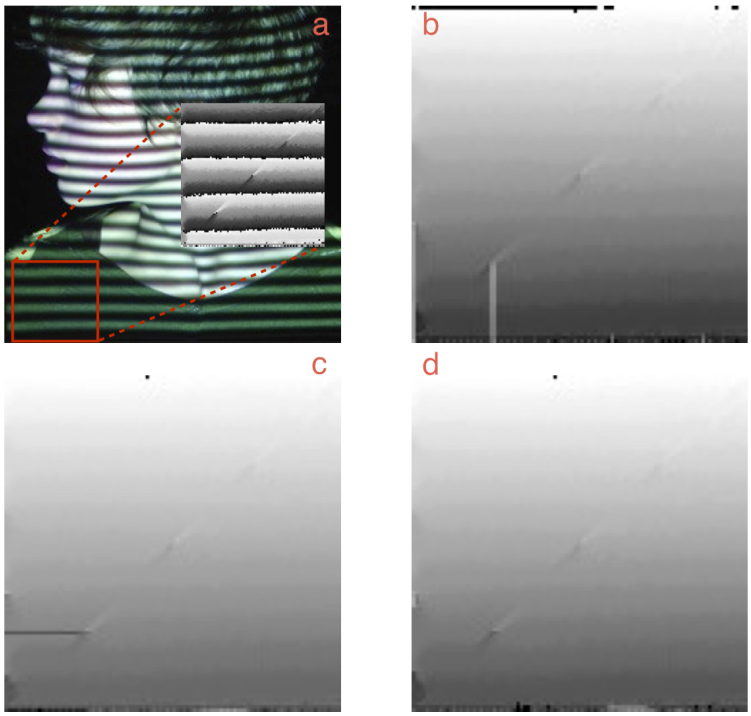


Figura 4.10: Phase unwrap guiado por prioridad

La figura 4.10 muestra también errores presentes en los límites de la imagen. Esto se debe a que bordes de la imagen presentan deformaciones y son excluidas del algoritmo de calidad, de forma que además suelen quedar para ser procesadas en último lugar con un valor muy alto en la corrección *wrap*. Una posible mejora para el sistema podría ser eliminar completamente la franja próxima al borde tanto a la hora de calcular calidades como al procesar *wrap* y *unwrap*.

Sin embargo, la evaluación guiada por calidad no consigue resultados perfectos teniendo sus propios vicios que han de ser depurados. El comportamiento general del algoritmo es el de procesar franjas enteras antes de continuar con otra franja. La tendencia a encontrar píxeles con buena puntuación hace que zonas poco o nada relevantes se procesen con alta prioridad, alejando el foco de interés del algoritmo del área anteriormente procesada. De esta forma discontinuidades que deben ser eliminadas se conservan como si se tratara de discontinuidades naturales de la imagen.

El algoritmo guiado por prioridad no soluciona correctamente el problema como se puede ver en la imagen 4.11. Existen discontinuidades que han sido evaluadas correctamente con resultados muy buenos, sin embargo muchas de estas discontinuidades han sido ignoradas:

“saltadas“ mediante algún artificio producido por áreas en los límites de la imagen con valores de calidad erróneos.

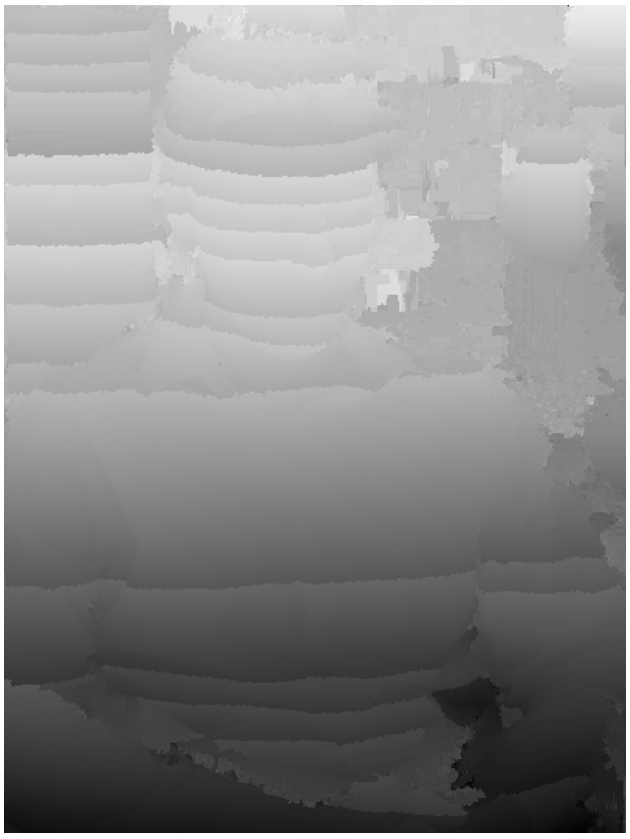


Figura 4.11: error en *phase unwrap* guiado por prioridad

Otra consideración importante al usar este algoritmo es que resulta significativamente más costoso que los anteriores. A pesar que la complejidad de éste sigue siendo lineal (solamente se procesa un píxel una única vez), en realidad la cola de prioridad añade un extra con la ordenación, que a pesar de utilizar algoritmos de buen rendimiento, con imágenes grandes resulta muy poco provechoso. Hay que recordar que cada píxel procesado añade cuatro vecinos a la cola por lo cual el tamaño de ésta crece exponencialmente y resulta necesario ordenar cada uno de éstos para que sea evaluado en el instante correcto.

Capítulo 5

Resultados

Una vez descrita la técnica three phase shift y las etapas que la componen se procede a explicar los resultados obtenidos a raíz de los experimentos realizados.

5.1. Resultados con respecto a *phase wrap*

La operación *phase wrap* es una operación que resulta costosa en cómputo. Un gran trabajo por parte de la comunidad científica se ha llevado a cabo sobre *three phase shift* para hacer la operación *phase wrap* menos relevante o para substituirla por otra diferente, es por esto que los esfuerzos de este proyecto se han centrado en analizar las posibles mejoras que se pueden aplicar en este área y de esta manera ayudar a la etapa *unwrap*. Los experimentos ejecutados muestran que el filtro bilateral no es tan productivo como cabía esperar, como se puede observar en la figura 5.1 la ejecución del phase *unwrap* con prioridad antes y después de aplicar el filtro resulta en los mismos errores, dirigidos por la evaluación de la calidad.

El la figura 5.1 en la imagen de arriba a la izquierda se encuentra el mapa *wrap* original, mientras que arriba a la derecha se encuentra el resultado de aplicar el filtro bilateral. Abajo se muestra el resultado tras aplicar *phase unwrap* con el algoritmo de inundación guiado por calidad a ambos mapas. Se puede observar que las dos imágenes son aparentemente iguales, es necesario mencionar que muestran diferente suma de comprobación. Sin embargo en los dos casos se repiten los mismos errores, "saltando"discontinuidades.

5.1.1. Características del mapa *wrap* que lo hacen inadecuado para el uso del filtro bilateral

Como se ha explicado anteriormente el mapa *wrap* es el resultante de fusionar las tres imágenes originales cada una con un desfase en el patrón. Como revelan los experimentos realizados, este mapa tiene características particulares que lo hacen poco adecuado para la utilización de un filtro bilateral. En la imagen resultante no existen otras discontinuidades que las que producen un salto por alcanzar una cota ($-\pi$ o π), o las discontinuidades propias

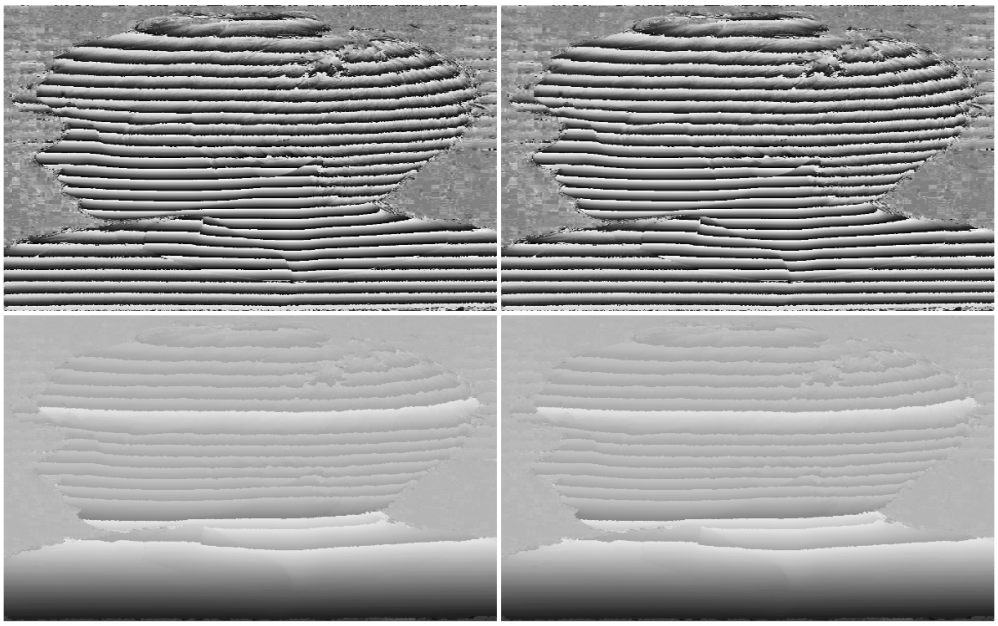


Figura 5.1: *Phase unwrap* con y sin proceso bilateral

de la imagen, esto es: los límites entre objetos. El resto de la imagen tiene franjas en las cuales existe una transición suave entre la cota mínima ($-\pi$) y la cota máxima (π). El filtro bilateral no realiza modificaciones relevantes en el comportamiento del mapa debido a la transición suave entre valores que se produce dentro de las franjas continuas del mapa *wrap*. El interés principal de realizar un filtrado sobre este mapa es la de eliminar bordes dentados y picos puntuales que alteren el comportamiento del algoritmo *phase unwrap*.
Este filtro bilateral no reduce los picos puntuales considerablemente ya que se elimina el aporte del área circundante. Los bordes dentados sufren un efecto similar. Finalmente no se consigue un efecto relevante mediante el uso de este filtro, por lo que para mejorar el trabajo de la etapa *phase unwrap* es necesaria la intervención de algún procesado local, quizá guiado por un detector de bordes como el filtro laplaciano o sobel.

5.2. Consideraciones de rendimiento

El rendimiento de los algoritmos es algo que conviene evaluar detenidamente. Como ya se ha comentado anteriormente la mayoría de los esfuerzos se centran en acelerar la captura, mejorando el rendimiento de la infraestructura, o acelerar las etapas *phase wrap* y *phase unwrap*. Sin embargo en este trabajo se ha insertado una etapa intermedia de filtrado con el objetivo de mejorar la calidad de la salida, sin tener en cuenta la repercusión en el rendimiento global de la aplicación.

La tabla 5.1 corresponde a las pruebas realizadas se han llevado a cabo con un Intel Core 2 Duo a 2.4GHz con 4 GB de memoria DDR3.

Cuadro 5.1: Tiempos de ejecución

	100x100	480x640
phase *wrap*	10.192ms	66.979ms
bilateral	41.989ms	1276.22ms
unwrap TopDown	0.498ms	18.263ms
unwrap inundación	2.344ms	65.175ms
unwrap Prioridad	104.065ms	4100.31ms

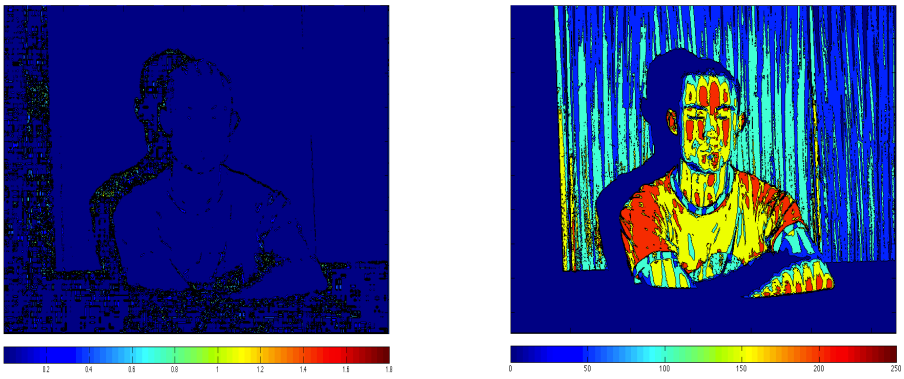
Como se observa en la tabla 5.1 la ejecución del filtro bilateral no es para nada despreciable. Los tiempos de ejecución son mayores que la etapa *wrap* y mayores que las dos primeras implementaciones de la etapa *unwrap*. A pesar de que el algoritmo se ha implementado sin hacer uso de optimización alguna, los datos demuestran que es demasiado costoso para ser insertado como una etapa más en *three phase shift*.
También es necesario destacar el relevante incremento de tiempo que se produce en la etapa *unwrap* cuando se utiliza el algoritmo guiado por prioridad frente a los dos algoritmos anteriores. Esto se debe al coste extra de mantener la cola de prioridad ordenada, además es necesario mencionar que este algoritmo no escala bien, ya que al añadir a la cola 4 elementos por píxel y despues extraer uno, el crecimiento es exponencial, por lo que en cada iteración es posible que sea necesario realizar más comprobaciones para insertar un elemento en una posición correcta.
Por último, cabe remarcar que la implementación de McDonald and Weber [15] realiza la etapa *unwrap* con resultados aceptables en 22.26ms para un tamaño 480x640, haciendo uso del algoritmo por inundación de Zhang et al. [23]. No obstante en este algoritmo no toda la imagen es evaluada sino que en una etapa anterior se enmascaran píxeles que no serán procesados por no alcanzar un umbral mínimo de calidad. Esto repercute en una buena interpretación de la profundidad pero solo en las áreas que pasen este test de calidad, además, al tratarse de un algoritmo de inundación, muestra problemas para identificar bifurcaciones como se explica en el apartado 4.3.1.

5.3. Resultados con respecto a *phase unwrap*

Los algoritmos de recorrido son fácilmente implementables pero ineficaces ya que pueden dar con soluciones poco acertadas.
El algoritmo basado en prioridad escoge un camino para realizar la corrección de todo el mapa *wrap* con discontinuidades. El problema es que la función de calidad debe estar adecuadamente escogida para que este camino sea productivo y no produzca artificios.

A continuación se analizan los problemas encontrados durante el cálculo de la calidad de los píxeles y cómo éstos producen la evaluación de caminos incorrectos. Como se explica

en el apartado 4.3.1, la calidad se basa en dos parámetros: la distancia y el rango.
La figura 5.2a corresponde con un la evaluación de la distancia entre el valor escalar de intensidad del píxel y sus cuatro vecinos, norte, sur, este y oeste. En esta imagen se puede observar como las regiones azules corresponden a valores cercanos a 0, las cuales son la mayoría . El color negro tiene un valor 0, estas áreas corresponden a las sombras o zonas donde no se tiene ninguna información de fase, solamente la luz ambiente. Tambien se observan motas de color verde y, menos visibles, rojas, que corresponden a valores extremos donde los píxeles muestran grandes divergencias con sus vecinos. Éstas son las zonas menos deseadas como referencia a la hora de corregir los píxeles vecinos.



(a) Distancia (b) Rango

Figura 5.2: Mapas de distancia y rango entre píxeles para la evaluación de calidad

En la siguiente figura 5.2b se puede observar el resultado de calcular el factor rango, que corresponde al rango en el cual se encuentran los valores de las tres imágenes originales. Es necesario mencionar que este parámetro es calculado sobre la imagen en escala de grises de 8 bits, por lo que el rango máximo es 255. En la figura se puede ver que el valor de rango no sobrepasa 250 para ningún píxel. Esas zonas con valores máximos son las ideales para empezar y serán las mejores candidatas para extraer de la cola durante la ejecución del algoritmo. Estas zonas con valor máximo son aquellas en las cuales una de las fases es mínima y otra de las tres máxima, teniendo de esta manera máxima información de profundidad.

Por último, la figura 5.3 muestra el mapa de calidad para todos los píxeles, éste corresponde a la distancia dividida por el rango. Se puede observar dos características que hacen la evaluación de la imagen errónea. Primeramente están los valores en blanco y rojo, como se puede observar en la escala de la leyenda estos valores no corresponden a resultados reales. En el caso de que tanto la distancia como el rango sean cero, entonces se computa *Nan* (*not a number*, mostrado en blanco) mientras en el caso de que solo el divisor sea cero, con valor infinito (mostrado con color rojo). Esto último no es excesivamente negativo, ya que podemos filtrar fácilmente estos valores para que sean evaluados en último lugar, penalizan-

do su puntuación de calidad con un valor alto (el infinito es un valor válido). Sin embargo, las regiones que sí son preocupantes son aquellas que, perteneciendo a una sombra, en la cual no hay información de fase y solamente existe información de luz ambiental, pequeñas fluctuaciones de intensidad entre tomas consiguen que exista un valor de calidad válido para el píxel, y lo que es peor, que este sea cercano a cero, por lo que es un valor de alta prioridad.

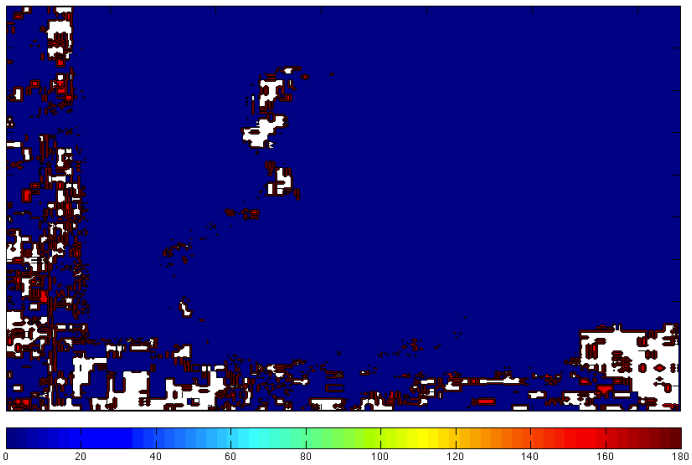


Figura 5.3: Mapa de calidad

El efecto anteriormente explicado hace que el algoritmo basado en prioridad "salte" fuera del área de datos válidos para evaluar una zona de sombra, en la cual no existe discontinuidad alguna a excepción de las divisiones por cero. Tras haber evaluado este área, el algoritmo vuelve a evaluar la zona de franjas con el píxel mejor puntuado. Este píxel puede estar en cualquier posición y separado cualquier número de saltos de la última franja evaluada, por lo que no garantiza el "desenrollado" correcto de todas las franjas. Este comportamiento es válido también para la figura 4.10 en la sección 4.3.2, ya que ésta tiene franjas en su totalidad, sin tener áreas de sombras.
Este problema se ve producido porque los algoritmos utilizados solamente tienen en cuenta uno de los vecinos de un píxel para realizar la corrección, siendo muy propensos a transmitir ruidos. Una solución en la que cada píxel se corrige teniendo en cuenta los píxeles vecinos que ya han sido corregidos podría ser buena solución, pero resultaría costoso como ya se ha visto anteriormente.

Capítulo 6

Conclusión y trabajo futuro

Finalmente para concluir este trabajo de investigación se exponen las consideraciones a tener en cuenta a la hora de trabajar con la técnica *three phase shift*

6.1. Conclusión

La técnica *Three Phase Shift* demuestra ser un método muy eficaz para obtener la estructura tridimensional de una escena u objeto. La complejidad de este método radica en una correcta implementación de los algoritmos escogidos, frente al uso de dispositivos complicados de otras técnicas, como por ejemplo el tiempo de vuelo.
Por otra parte, *phase unwrap* es una técnica utilizable en otros campos, como en la corrección de las imágenes médicas mediante resonancia magnética (MRI[1]). Los esfuerzos realizados para mejorar el rendimiento de esta costosa operación o para mejorar el comportamiento de ésta pueden ser fácilmente aplicados a otros campos de investigación.

6.2. Materiales

Para hacer una infraestructura de escaneado mediante *three phase shift* es necesario un proyector y una cámara como se puede ver en la figura 6.1. En este caso se ha trabajado con un proyector Acer X110 de tecnología DLP y se han probado varias cámaras, una webcam firewire Unibrain Fire-i y una cámara Olimpus c-5050.

El sensor de la webcam con la que se ha trabajado es pequeño y por otra parte el proyector de tipo DLP tiene una gran potencia para su modesto precio. Esto repercute en que las imágenes resultantes se encontraban constantemente sobreexpuestas. Por lo tanto se opto por prescindir de la cámara Firewire y realizar las pruebas con una cámara de fotos convencional.
Con la cámara fotográfica se consiguen buenos resultados, pero resulta más complicado de sincronizar con el sistema, en este caso fue necesario hacerlo de forma manual: tomando las fotografías y cambiando la fase. Posteriormente las fotografías resultantes tenían un gran

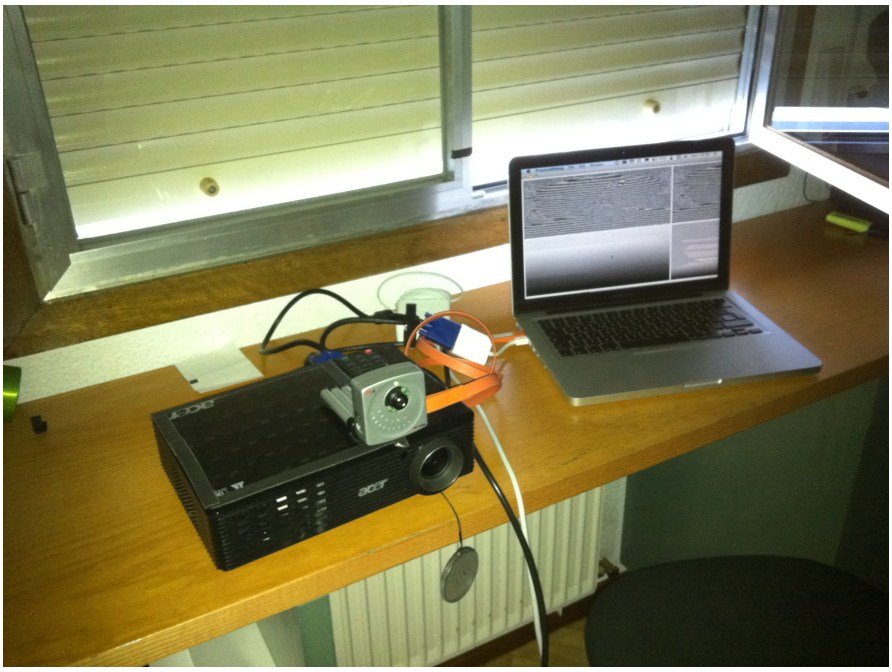


Figura 6.1: Infraestructura para la captura *three phase shift*

tamaño, con 2560x1920 píxeles. El proyector tiene una resolución de 800x600 píxeles por lo que fue necesario reducir las capturas a este tamaño, ya que no es posible que tengan más datos de fase de los que el proyector sea capaz de inferir.

Tiempo real

Para realizar capturas en tiempo real y obtener vídeo 3D es necesario precisar un poco más con la infraestructura que es necesaria. Los dispositivos DLP para proyectores hacen uso de espejos para reflejar el haz de luz para cada uno de los píxeles, éstos se activan o desactivan varias miles de veces por segundo y son capaces de reflejar la luz con diferente intensidad. Para añadir color a la imagen proyectada, el haz de luz original pasa a través de un disco con los colores básicos, que tiñe de determinado color el haz de luz y simultáneamente se activan los espejos de los píxeles que tengan este color en alguna de sus componentes. Este disco puede hacer que la toma de imágenes a alta velocidad produzca la aparición de colores distintos al gris, y arruine de esta manera el modo de adquirir la profundidad.
Para este tipo de usos, la captura de vídeo 3D, es necesario utilizar un proyector LCD u otra tecnología que no provoque artificios. También es posible el uso de proyectores DLP mediante su manipulación previa (es necesario eliminar el disco de color) y la utilización de cámaras de alta velocidad como realiza Huang Et al. en su trabajo[13].

6.3. Trabajo futuro

Durante la realización de este proyecto, han surgido ciertas cuestiones que han quedado fuera del alcance del proyecto original pero que pueden ser abordadas en una investigación futura.

- analizar filtros en otros dominios, por ejemplo en frecuencia.
- más algoritmos para implementar la etapa *phase unwrap*.
- estudiar el uso de proyectores de otra tecnología y cámaras de altas prestaciones.

Sin embargo, las dos áreas principales que cabe destacar se indican a continuación.

6.3.1. Real-time framework

Una de las aplicaciones más interesantes que encuentro para esta técnica es la de extraer información de un entorno tridimensional en tiempo real. Esto podría conseguirse gracias a un pipeline de software que realizara, por cada tres imágenes, el análisis de profundidad.
Existen dos aspectos principales a estudiar para el correcto funcionamiento del framework:

Utilización incremetal de los originales

Para la adquisición de la profundidad es necesario analizar 3 imágenes, cada una con el patrón desfasado respecto del patrón anterior. El trabajo consiste en analizar si es posible reutilizar los instantes de tiempo t_{-1} y t_{-2} y el fotograma del instante de tiempo actual t_0. Si esto fuera posible, se podría calcular un mapa de profundidad por cada captura, en lugar de uno cada tres capturas.
Es necesario mencionar también que la escena puede corresponder a un entorno dinámico, en el cual existen piezas u objetos en movimiento. Obviamente las tres fotografías no pueden corresponder al mismo instante pero, de todas formas, se puede conseguir información general sobre la posición relativa del objeto dentro del rango de tiempo que lleva tomar las tres imágenes.
El uso de esta modificación, traería problemas así mismo: El mapa de profundidad es relativo y depende de la posición de las fases proyectadas. En este caso, cada tres salidas conseguidas se cierra el ciclo, el cual comienza con, por ejemplo: con un cambio de fase $\phi - 120°$ el segundo comienza con ϕ y el tercero $\phi + 120°$. Los resultados obtenidos no serían iguales puesto que el punto de partida varía.

Utilización de la información del anterior fotograma

Para poder hacer realidad lo propuesto anteriormente sería posible utilizar la información del último proceso para, de esta manera, ayudar al siguiente. El mapa de profundidad

conseguido no puede diferir mucho del anterior y la geometría debería ser similar. Debido al paso del tiempo los objetos pueden sufrir algún tipo de transformación. Se debe estudiar si esta transformación puede ser corregida por la información procedente de la imagen con fase del anterior instante.

6.3.2. GPGPU

Un área de investigación que podría resultar muy provechosa es la implementación del método haciendo uso de aceleradores vectoriales. En este caso, procesadores masivamente paralelos como las GPUs. El uso de GPU para cómputo general es una tendencia actual y consigue muy buenos resultados en el procesado de grandes volúmenes de datos como en el caso del tratamiento imágenes.
Los trabajos analizados sobre Three Phase Shift siempre cuentan con un enfoque hacia el rendimiento del sistema, y ya se ha utilizado esta tecnología en algunos artículos recientes. Sin embargo existe poco desarrollo sobre esto, sobre todo en lo que a *phase unwrap* concierne.

Phase Wrap

Como ya se menciona en el apartado 3.2.1, la implementación de la etapa *unwrap* resulta inmediata, dado que la naturaleza del problema es trivialmente paralelizable. Sin embargo es necesario realizar un pequeño estudio acerca de las implicaciones del uso de ciertas operaciones:
Arctan es una operación muy costosa en CPU, pero en los multiprocesadores de una GPU puede serlo aún más. En CPU esto se solventa haciendo uso de memoria extra para construir una tabla LUT, donde se puede calcular de antemano los resultados de *arctan*.
Sin embargo en GPU el uso de memoria extra es contraproducente, ya que es ahí donde se encuentra su mayor cuello de botella. Es necesario evaluar el rendimiento de una tabla LUT (la cual podría estar implementada en la memoria reservada para texturas), frente al del uso de *arctan* para cada píxel.

Phase Unwrap

Phase Unwrap es un algoritmo mucho más interesante de paralelizar. Las primitivas implementaciones contempladas en el apartado 4.2 muestran una serie de transformaciones iterativas de carácter global y otras que operan por regiones. La implementación paralela de éstas podría llevarse a cabo mediante la operación simultánea de elementos independientes. En el caso de los algoritmos globales, esto supone evaluar de forma paralela cada uno de los píxeles dentro de la misma iteración del algoritmo. En el caso de los algoritmos de corrección por regiones es posible evaluar las distintas regiones simultáneamente, y posteriormente corregir éstas dos a dos, terminando la ejecución del algoritmo mediante un árbol binario.

Por otra parte, el interés principal es la realización de forma paralela de los algoritmos de recorrido. Los algoritmos basados en recorrido tienen mejor rendimiento a pesar de ser más vulnerables a ruidos, es por este motivo que creo conveniente aplicar nuevas técnicas mediante las cuales se pueda conseguir un mejor comportamiento y rendimiento.
Implementar la solución basada en recorrido para su ejecución en GPU podría seguir las siguientes lineas. Primeramente se evalúa un recorrido, el cual puede ser escogido de forma arbitraria, o bien basado en función de la calidad como se explica en el apartado 4.3.1. Posteriormente se divide este recorrido en secciones y se procesan simultáneamente, realizando reconstrucciones parciales a partir del primer píxel.
A continuación es necesario corregir el desplazamiento de cada elemento con respecto al segmento anterior. Para evitar que esto se convierta en un proceso iterativo con un número de iteraciones igual al número total de discontinuidades corregidas, es necesario computar el desplazamiento acumulado de cada una de las secciones anteriores y añadirlo a cada píxel. Esto se puede realizar mediante una operación scan como por ejemplo explica Harris[10] en su manual sobre la implementación de dicha operación en GPU.

Bibliografía

[1] Agosto 2011. URL http://en.wikipedia.org/wiki/Magnetic_resonance_imaging.

[2] Agosto 2011. URL http://www.mathworks.es/help/toolbox/images/ref/fspecial.html#bqkft1o.

[3] Agosto 2011. URL http://es.wikipedia.org/wiki/Z-Buffer.

[4] Agosto 2011. URL http://en.wikipedia.org/wiki/Gray_code.

[5] Agosto 2011. URL http://www.leica-geosystems.com/en/Leica-System-1200-Leica-TPS1200_4547.htm.

[6] Agosto 2011. URL http://es.wikipedia.org/wiki/Lookup_table.

[7] Agosto 2011. URL http://openkinect.org/wiki/Main_Page.

[8] Agosto 2011. URL http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/.

[9] L.-S. Bieri and J. Jacot. Three-dimensional vision using structured light applied to quality control in production line. *Ecole Polytechnique Federale de Lausanne*, 2004.

[10] Mark Harris. *Parallel Prefix Sum (Scan) with CUDA*. Nvidia, April 2007.

[11] Miguel Arevallilo Hernandez, David R. Burton, Michael J. Lalor, and Munther A. Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Applied Optics*, 41(35):37–44, Diciembre 2002.

[12] Peisen S. Huang and Song Zhang. Fast three-step phase-shifting algorithm. *Applied Optics*, 45(21):87–91, Julio 2006.

[13] Peisen S. Huang, Chengping Zhang, and Fu-Pen Chiang. High-speed 3-d shape measurement based on digital fringe projection. *Opt. Eng.*, 163 (2003); doi:10.1117/1.1525272, 2003.

[14] Gonzalo Pajares Martinsanz and Jesus M. de la Cruz García. *Vision por Computador*. Ra-Ma, 2007.

[15] Kyle McDonald and Gunter Weber. Structured light 3d scanning, 2011. URL http://sites.google.com/site/structuredlight/techniques.

[16] Srinivasa G. Narasimhan, Sanjeev J. Koppal, and Shuntaro Yamazaki. Temporal dithering of illumination for fast active vision. *European Conference on Computer Vision (ECCV)*, October 2008.

[17] J. Reznicek and K. Pavelka. New low-cost 3d scanning techniques for cultural heritage documentation. *Remote Sensing and Spatial Information Sciences*, 37(B5), 2008.

[18] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *IEEE International Conference on Computer Vision*, 1998.

[19] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, Zhiguo Li2, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. *EUROGRAPHICS*, 2004.

[20] Yongchang Wang. *Novel Approaches in Structured Light Illumination*. PhD thesis, University of Kentucky, 2010.

[21] Song Zhang and Peisen S. Huang. High-resolution, real-time three-dimensional shape measurement. *Optical Engineering*, 2006.

[22] Song Zhang and Shing-Tung Yau. High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. *Optical Engineering*, 46(11), 2007.

[23] Song Zhang, Xiaolin Li, and Shing-Tung Yau. Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction. *Applied Optics*, Vol. 46:50–57, 2007.

Índice de figuras

1.1. Visión estereoscópica 2
1.2. Análisis de la Visión estereoscópica 3
1.3. Sensor de tiempo de vuelo 4
1.4. Sensor comercial luz estructurada 5

2.1. Estructura de three phase shift 8
2.2. Patrones desfasados 9
2.3. Error en el proceso 12
2.4. Error en el proceso 13

3.1. Imágenes Phase Wrap 14
3.2. Ejemplo de una posible implementación en GPU usando CUDA 17
3.3. Mapa wrap antes de filtrar 18
3.4. Filtrado Motion 20
3.5. Filtrado Laplaciano 21

4.1. Operación de phase unwrap 24
4.2. Código C de la operación básica de unwrap 24
4.3. dirección de unwrap 25
4.4. Mapa wrap para los experimentos 27
4.5. UnWrap Top-down 28
4.6. UnWrap Inundación 29
4.7. Discontinuidad producida por el recorrido 29
4.8. Comportamiento Unwrap 30
4.9. Cómputo de la calidad del píxel 30
4.10. Phase Unwrap por prioridad 32
4.11. Phase Unwrap por prioridad erroneo 33

5.1. Resultados filtrado bilateral 35
5.2. Mapa de distancia 37
5.3. Mapa de calidad 38

6.1. Infraestructura utilizada 40

A.1. Diagrama de flujo Phase unwrap Top-down 49
A.2. Diagrama de flujo Phase unwrap con cola 50

Apéndice A

Diagramas de ejecución

A.1. *phase unwrap* Top-down

En la figura A.1 se puede ver el comportamiento del algoritmo top-down explicado en el apartado 4.3.

El algoritmo top-down trata, con total independencia las columnas de la imagen, realizando un tratamiento unidimensional. Todos los píxeles de la imagen a excepción de los pertenecientes a la última columna son utilizados para la evaluación de algún pixel, lo cual hace que los errores se propaguen drásticamente. Es necesario mencionar que el algoritmo funciona de forma vertical de la misma manera hacia arriba o abajo, pero en vertical siempre que la captura se haya realizado con patrones horizontales. Para el uso de patrones verticales es necesario trasponer el comportamiento de este algoritmo.

A.2. *phase unwrap* por inundación

El la figura A.2 se muestra el comportamiento de los algoritmos basados en cola explicados en 4.3 y 4.3.2.

Los algoritmos por inundación no son más que una forma de realizar un recorrido unidimensional de un espacio bidimensional. El inconveniente es que se realiza la operación de corrección con respecto a un único valor, perdiendo parte de la información en pos de un incremento del rendimiento. Para las versiones guiadas por calidad solo es necesario ordenar la cola en función de ésta.

Una característica de este algoritmo es que un píxel puede estar dentro de la cola en varias formas dependiendo de sus predecesores, esto hace que el consumo de memoria se dispare, siendo necesario almacenar más píxeles que el tamaño original de la imagen, e incrementa el tiempo debido a la ordenación en los algoritmos guiados por calidad.

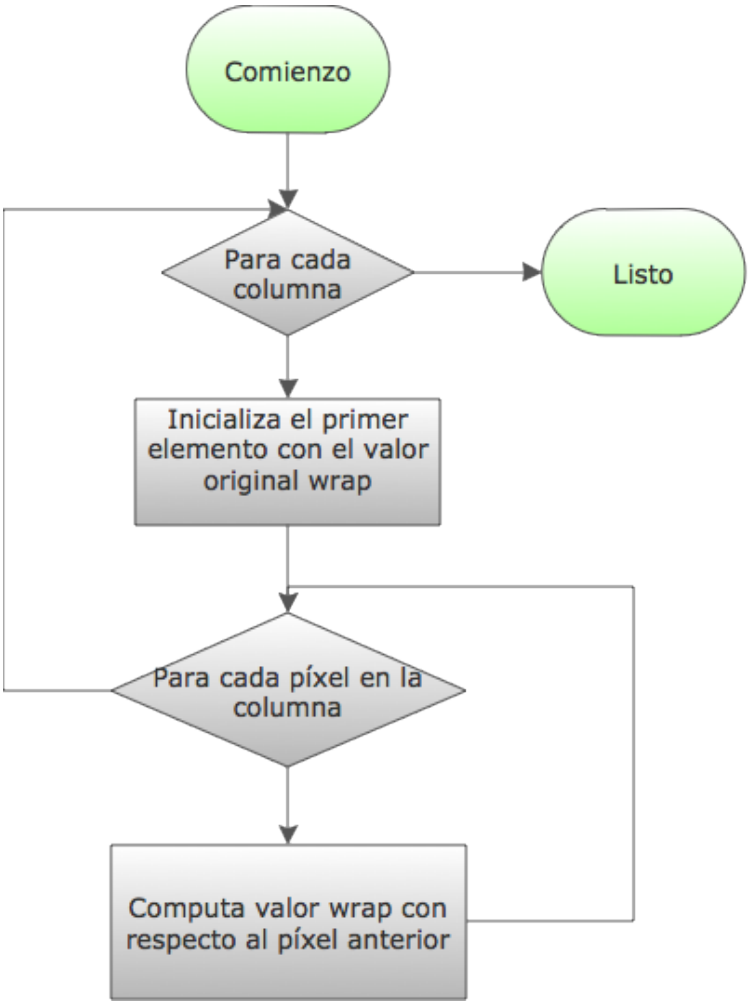


Figura A.1: Diagrama de flujo *phase unwrap* Top-down

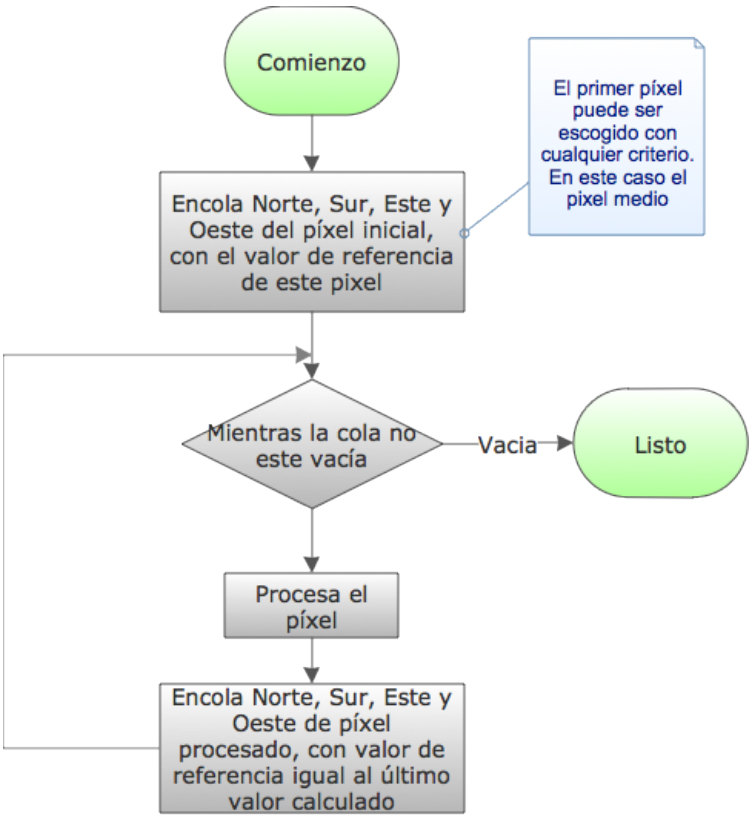


Figura A.2: Diagrama de flujo *phase unwrap* por inundación